**##1: Set up the REST API**

**1.Install Dependencies**

mkdir fabric-rest-api

cd fabric-rest-api

npm init -y

npm install express fabric-network dotenv

**2.Create the API:**

**##server.js**

const express = require('express');

const app = express();

const port = process.env.PORT || 3000;

const fabricHelper = require('./fabricHelper');

app.use(express.json());

// Endpoint to query the transaction by Dealer ID

app.get('/transaction/:dealerId', async (req, res) => {

const dealerId = req.params.dealerId;

try {

const result = await fabricHelper.queryTransaction(dealerId);

res.status(200).json({ result });

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// Endpoint to create a new transaction

app.post('/transaction', async (req, res) => {

const { dealerId, msisdn, mpin, balance, status, transAmount, transType, remarks } = req.body;

try {

const result = await fabricHelper.createTransaction(dealerId, msisdn, mpin, balance, status, transAmount, transType, remarks);

res.status(200).json({ result });

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// Start server

app.listen(port, () => {

console.log(`Server is running on port ${port}`);

});

**##fabricHelper.js**

'use strict';

const { Gateway, Wallets } = require('fabric-network');

const path = require('path');

const fs = require('fs');

require('dotenv').config();

const ccpPath = path.resolve(\_\_dirname, process.env.CONNECTION\_PROFILE);

const ccp = JSON.parse(fs.readFileSync(ccpPath, 'utf8'));

// Load wallet path from the environment variable

const walletPath = path.join(process.cwd(), process.env.WALLET\_PATH);

async function connectToGateway() {

const wallet = await Wallets.newFileSystemWallet(walletPath);

const gateway = new Gateway();

await gateway.connect(ccp, {

wallet,

identity: process.env.IDENTITY,

discovery: { enabled: true, asLocalhost: true }

});

const network = await gateway.getNetwork(process.env.CHANNEL\_NAME);

const contract = network.getContract(process.env.CONTRACT\_NAME);

return { gateway, contract };

}

async function queryTransaction(dealerId) {

const { gateway, contract } = await connectToGateway();

const result = await contract.evaluateTransaction('queryTransaction', dealerId);

await gateway.disconnect();

return result.toString();

}

async function createTransaction(dealerId, msisdn, mpin, balance, status, transAmount, transType, remarks) {

const { gateway, contract } = await connectToGateway();

const result = await contract.submitTransaction('createTransaction', dealerId, msisdn, mpin, balance, status, transAmount, transType, remarks);

await gateway.disconnect();

return result.toString();

}

module.exports = { queryTransaction, createTransaction };

**###Create a .env file to store environment variables**

PORT=3000

CONNECTION\_PROFILE=fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/connection-org1.json

WALLET\_PATH=fabric-samples/test-network/wallet

IDENTITY=user1

CHANNEL\_NAME=mychannel

CONTRACT\_NAME=transactionContract

**###Test the REST API Locally**

cd fabric-samples/test-network

./network.sh up createChannel -c mychannel -ca

**###Run The API server:**

node server.js

**###Query A Transaction:**

curl http://localhost:3000/transaction/D1001

**###Create a Transaction:**

curl -X POST http://localhost:3000/transaction \

-H "Content-Type: application/json" \

-d '{"dealerId":"D1003","msisdn":"1122334455","mpin":"3333","balance":"500","status":"Active","transAmount":"200","transType":"Credit","remarks":"Test Transaction"}'

**### Create a Dockerfile:**

# Use the official Node.js image

FROM node:14

# Set the working directory

WORKDIR /usr/src/app

# Copy package.json and package-lock.json

COPY package\*.json ./

# Install dependencies

RUN npm install

# Copy the rest of the application

COPY . .

# Expose the API port

EXPOSE 3000

# Start the API server

CMD ["node", "server.js"]

**###Build the Docker Image**

docker build -t fabric-rest-api .

Command(fabric-rest-api) create the docker image.

###Run The Docker Container

docker run -d -p 3000:3000 --name fabric-rest-api-container fabric-rest-api

**###Clean UP**

docker stop fabric-rest-api-container

docker rm fabric-rest-api-container

**### Verify Smart Contract Logs**

docker logs peer0.org1.example.com