SenShaMart Project: User Manual

SenShaMart project is a JavaScript-based project, and it was built by using NodeJS V14.17.5. To run SenShaMart project on your machine(s), you need the following:

**Software Requirements:**

* *SenShaMart code:* you can get SenShaMart code form <to be add later>. Copy the code folder on your preferred directory.
* *A Command line application:* You can use the native Terminal application for both Linux and MacOS and native CMD application for Windows.
* *A code editor (Optional):* Use your preferred code editor. But if you are looking for a new one, I would recommend Visual Studio Code to edit the original code or the configuration file.
* *NodeJs:* it is an engine to run JavaScript-based projects. You can install it form this link <https://nodejs.org/en/download/>. Note that, you will need an appropriate installer for your system to be able to install it. After installing NodeJS, make sure it is installed properly by:
  + typing **$ node -v** in your command line application. As a result, you should get a response for something like **V14.17.5**.
  + typing **$ npm -v** in your command line application. As a result, you should get a response for something like 6.14.14.

open

* *Open SenShaMart directory and run the following commands*
  + *Npm run start*
* *Postman*: you need postman or a similar application (e.g., a web browser) to test the SenShaMart APIs by sending HTTP requests to the SenShaMart peer to peer servers. Here is the link to install Postman: <https://www.getpostman.com/docs/postman/launching_postman/installation_and_updates>
* *PM2 (optional):* it is a process manager for the JavaScript runtime Node.js. We use this tool to monitor the P2P servers in different terms like memory and CPU usage.

**Basic Commands**:

Note: all the following commands should run on the root of SenShaMart project directory.

* **$ Npm run start**

Run this command to start the first Peer-to-peer (WebSocket) server with default HTTP\_PORT = 3001, default P2P\_PORT =5001, and default MQTT\_PORT = 1884.

* **$ HTTP\_PORT=3\*\*\* P2P\_PORT=5\*\*\* MQTT\_PORT=18\*\* Npm run start**

Run this command to start the first Peer-to-peer (WebSocket) server with customised ports.

* **HTTP\_PORT=3002 P2P\_PORT=5002 MQTT\_PORT=1884 PEERS=ws://localhost:5\*\*\* npm run start**

Run this command to start the second Peer-to-peer (WebSocket) server with customised ports. The **PEERS=ws://localhost:5\*\*\*** Should contains the P2P\_PORT of the first peer.

* **HTTP\_PORT=3003 P2P\_PORT=5003 MQTT\_PORT=1885 PEERS=ws://localhost:50\*\*,ws://localhost:50\*\* npm run dev**

Run this command to start the third Peer-to-peer (WebSocket) server with customised ports. The **PEERS=ws://localhost:50\*\*,ws://localhost:50\*\*** Should contains the P2P\_PORT of the first and second peers.

* **You can add more peers by following the same procedure by giving the new peer suitable HTTP, P2P, and MQTT port numbers as well as adding the P2P\_ADDRESS of previous peers to PEERS.**
* **Npm run dev**

You can replace all the previous command with “**dev”** instead of “**start”** to do the same, but with nodemon.Nodemon is a tool that helps develop NodeJS based applications by automatically restarting the node application when file changes in the directory are detected.

* **Npm run test**

This command is used to run all tests we create to test SenShaMart functionalities.

* **pm2 start app/index.js**

this command is to start SenShaMart with a professional tool to monitor the behaviour of nodes.

* **pm2 start app/index.js --node-args="--max\_old\_space\_size=8192"**

the **--node-args="--max\_old\_space\_size=8192"** is used to increase the heap memory of the node. Note that you need to change the heap memory of each node individually.

**Essential Application Programming Interfaces (APIs):**

Note: to access any of these APIs, you need to know 1) the IP of the machine that is running as a server (peer), 2) **HTTP\_PORT**, and 3) the API name. For example: hrrp://136.186.108.192:3002/gen-key

/ChainServer/connect

/gen-key

/PubKeyFor

/chain-length

/Payment/Register

/Integration/All

/Integration/Register

/Integration/UsesOwnedBy

/Integration/OwnedBy

/Integration/OurBrokersBrokering

/Integration/OurBrokersWitnessing

/BrokerRegistration/All

/BrokerRegistration/Register

/BrokerRegistration/OwnedBy

/SensorRegistration/All

/SensorRegistration/Register

/SensorRegistration/OwnedBy

/sparql