# Besu-Box¶

To Be Used as Truffle Box Creating API Endpoints for Hyperledger Besu network.

We use a private blockchain for Ethereum BlockChain Development. This personalised blockchain is made with HL Besu Client.

The Smart Contract Written in solidity language is deployed on this Ethereum Permissioned Blockchain. Smart Contract is Immutable hence, once deployed it can't be changed.

A truffle box to serve as the foundation of any Truffle and Express.js dApp.

This Box Uses NodeJS(Express JS) to provide API endpoints to the Ethereum Blockchain smart contract so that this smart contract can be used in Android/iOS Apps as well.

Pre-Requisites 1.NodeJS 2.NestJS 3.Docker 4.Truffle 5.Besu Docker Image 6.Curl

Installation 1. Install Truffle and Nestjs globally

npm install -g truffle npm install -g @nestjs/cli

1. Download the box. This also takes care of installing the necessary dependencies.
2. truffle unbox illuzzig/besu-box
3. For quick, temporary tests this guide uses /tmp/besu/dev/ as mount volumes. Make sure you create the folders first in the root dir
4. mkdir -p /tmp/besu/dev/
5. To run a node that mines blocks at a rate suitable for testing purposes
6. // in another terminal (i.e. not in the truffle develop prompt)
7. // ensure you are inside the app directory when running this
8. npm
9. run
10. besu
11. :
12. docker
13. Now you can deploy your smart contracts.
14. // in another terminal (i.e. not in the truffle develop prompt)
15. // ensure you are inside the app directory when running this
16. truffle
17. migrate
18. --
19. network
20. besu
21. To run the Nestjs server
22. // in another terminal (i.e. not in the truffle develop prompt)
23. // ensure you are inside the app directory when running this
24. npm
25. run
26. start
27. :
28. dev
29. In a window terminal type
30. // in another terminal (i.e. not in the truffle develop prompt)
31. // ensure you are inside the app directory when running this
32. curl http://localhost:3000/balance/0xFE3B557E8Fb62b89F4916B721be55cEb828dBd73
33. As you can see this address holds all the metaCoin tokens accordin to the business logic implemented into the smart contract. Below the response{"address":"0xFE3B557E8Fb62b89F4916B721be55cEb828dBd73","balance":"10000"}
34. Set the variablemetaCoinAddress
35. (client_script/utils.js) to match the deployed MetaCoin address fromtruffle migrate
36. . You can get the smart contract address by typing
37. // in another terminal (i.e. not in the truffle develop prompt)
38. // ensure you are inside the app directory when running this
39. truffle networks | grep -i metacoin
40. Launch the transfer script
41. // in another terminal (i.e. not in the truffle develop prompt)
42. // ensure you are inside the app directory when running this
43. npm
44. run
45. transfer
46. The second address will receive 10 tokens from the first one. In a window terminal type

47. // in another terminal (i.e. not in the truffle develop prompt)
48. // ensure you are inside the app directory when running this
49. curl http://localhost:3000/balance/0x627306090abaB3A6e1400e9345bC60c78a8BEf57
50. Below the response{"address":"0x627306090abaB3A6e1400e9345bC60c78a8BEf57","balance":"10"}
51. For web service monitoring and performance metrics you can enable the APM agent in the main.ts file and visualize the incoming requests with kibana.Read More

Contributors 1.Giuseppe Gaetano Illuzzi