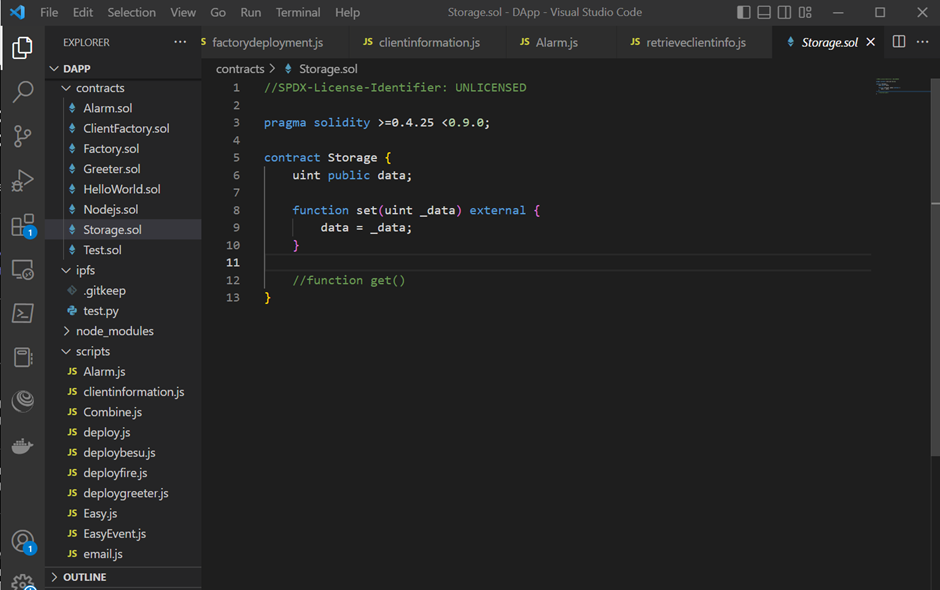
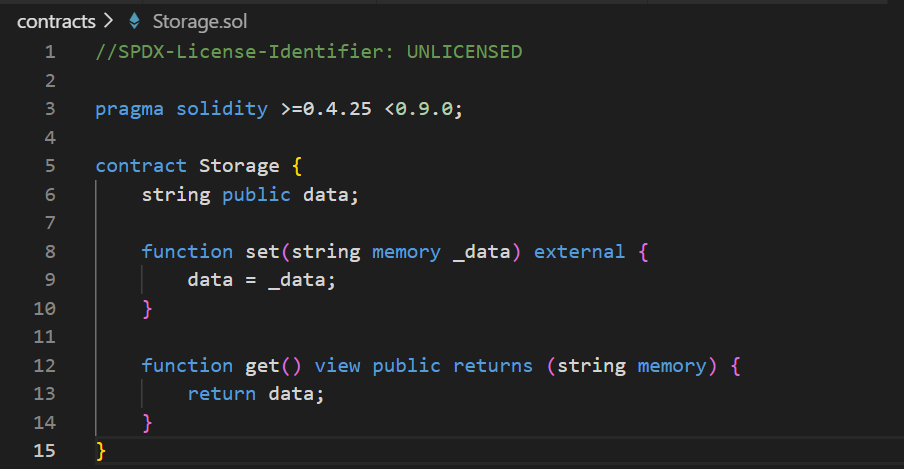
Deploying and Interacting with Test Smart Contract in Nodejs with Hardhat

* Create a new Nodejs directory and shift all the important scripts / solidity files from the Truffle Directory into that directory



* Update the Storage.sol Smart contract



* Download the Required Dependencies
  + Run npm init inside our directory on to command line to initialize our project with nodejs (if you haven’t)

npm init

* + Download Hardhat
    - Inside the project folder, run the following commands

npm install –save-dev hardhat

npm install --save-dev @nomiclabs/hardhat-ethers "ethers@^5.0.0"

* + - Create a Hardhat Project with an empty Hardhat.config.js

npx hardhat

You should then see a welcome message and option to select what you want to do. Select “create an empty hardhat.config.js”:

888 888 888 888 888

888 888 888 888 888

888 888 888 888 888

8888888888 8888b. 888d888 .d88888 88888b. 8888b. 888888

888 888 "88b 888P" d88" 888 888 "88b "88b 888

888 888 .d888888 888 888 888 888 888 .d888888 888

888 888 888 888 888 Y88b 888 888 888 888 888 Y88b.

888 888 "Y888888 888 "Y88888 888 888 "Y888888 "Y888

👷 Welcome to Hardhat v2.0.11 👷‍

What do you want to do? …

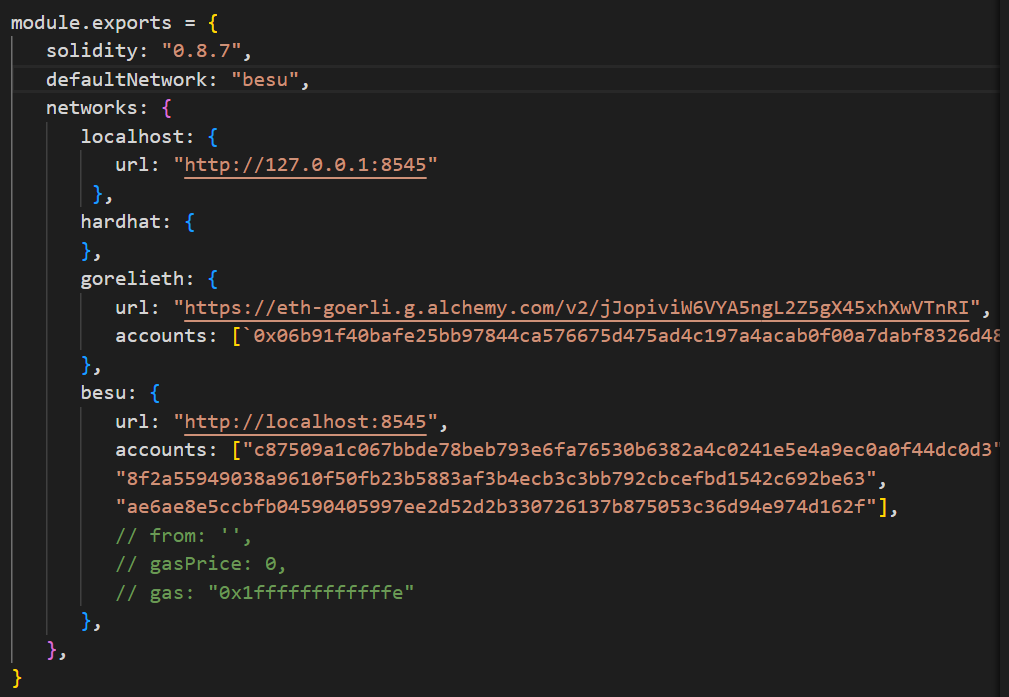
Create a sample project

❯ Create an empty hardhat.config.js

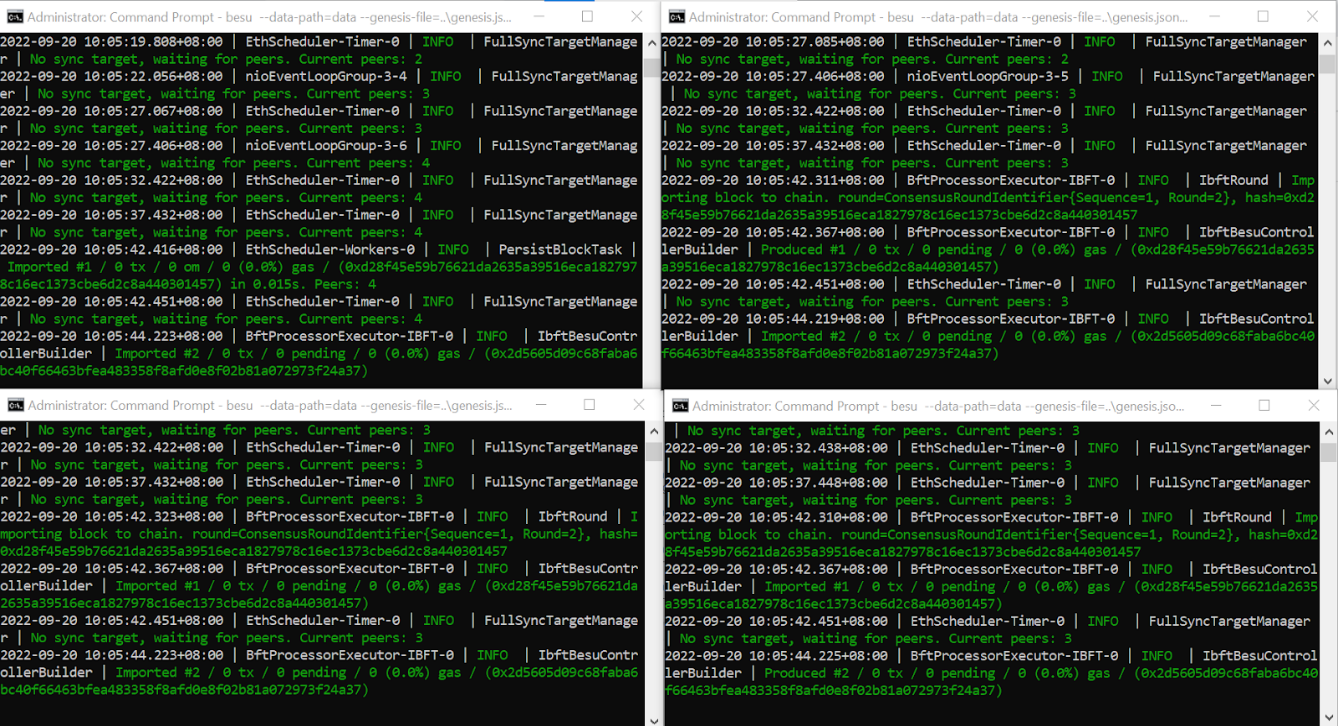
Quit

* Edit the Hardhat Config File to run Smart Contracts on our Private Blockchain Network

Input the following network besu:



* Deploy the Test Smart Contract
  + Run the Private Blockchain



* + Deploy the Test Smart Contract onto the besu network
    - Compile our Smart Contract

**npx hardhat compile**

* + - Write the Deploy Script

Under the Scripts Folder, deployStorage.js,

async function main() {

const Storage = await ethers.getContractFactory("Storage");

// Start deployment, returning a promise that resolves to a contract object

const storage= await Storage.deploy();

console.log("Contract deployed to address:", storage.address);

}

main()

.then(() => process.exit(0))

.catch(error => {

console.error(error);

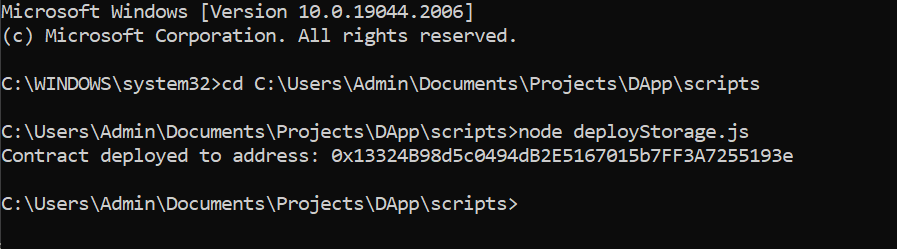
process.exit(1);

});

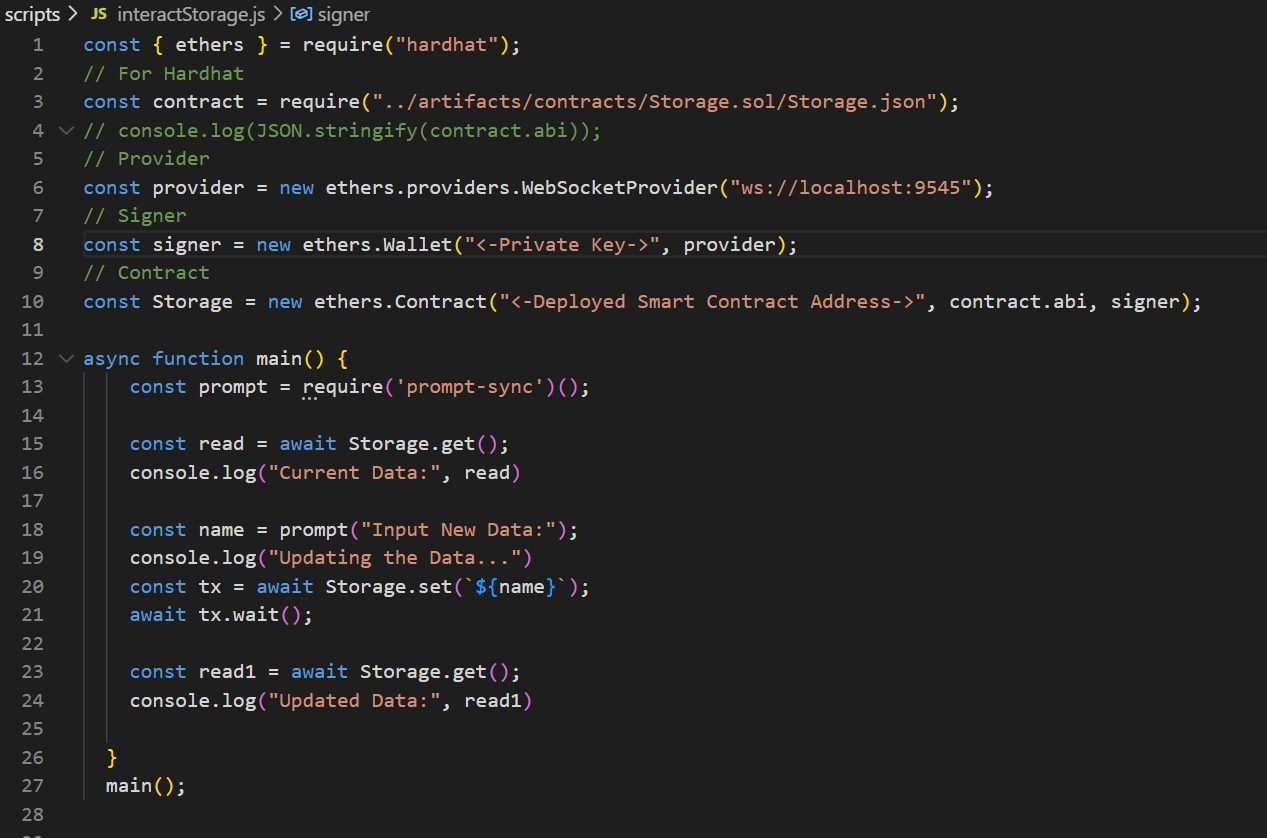
* + - Deploy the Test Contract

**node** **deployStorage**.js

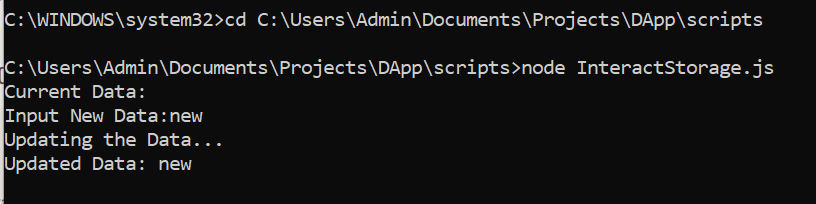
End Result:



* Interact with the Deployed Test Smart Contract with Hardhat and Nodejs
  + Grab the Test Smart Contract’s Abi, Provider, Signer and a Contract object for the specific contract deployed on chain
    - Create a Separate JavaScript File Called InteractStorage.js and input the following code:



* + Read and Update the Message
    - Run the InteractStorage.js File



Reference Links:

<https://www.web3.university/tracks/create-a-smart-contract/deploy-your-first-smart-contract>

<https://www.web3.university/tracks/create-a-smart-contract/interact-with-your-smart-contract>