Exploring the Template

The directory resembles a standard hardhat project. If we look into the contracts folder, we can see an existing example FHE contractcalculator.sol

. âââ artifacts âââ cache âââ contracts â âââ counter.sol

...

âââ hardhat.config.ts âââ package.json âââ README.md âââ tsconfig.json Let's use this contract to get familiar with the provided template.

If you haven't already, start your localfhenix instance by issuing the following command:

- npm
- yarn
- pnpm

npm run localfhenix:start yarn run hardhat localfhenix:start pnpm run localfhenix:start Now let's connect our wallet to our local test network. If you are using Metamask then open it up, click on the network dropdown button and add the localfhenix network.

Adding network details for localfhenix. Let's now explore the predefined hardhat tasks, run:

- npm
- yarn
- pnpm

npx hardhat yarn hardhat pnpm hardhat You will see the following:

AVAILABLE TASKS:

check Check whatever you need clean Clears the cache and deletes all artifacts compile Compiles the entire project, building all artifacts console Opens a hardhat console coverage Generates a code coverage report for tests deploy Deploy contracts etherscan-verify submit contract source code to etherscan export export contract deployment of the specified network into one file export-artifacts flatten Flattens and prints contracts and their dependencies. If no file is passed, all the contracts in the project will be flattened. gas-reporter:merge help Prints this message localfhenix:start Starts a LocalFhenix node localfhenix:stop Stops a LocalFhenix node node Starts a JSON-RPC server on top of Hardhat EVM run Runs a user-defined script after compiling the project sourcify submit contract source code to sourcify (https://sourcify.dev) task:addCount task:fhenix:usefaucet Fund an account from the faucet task:getCount task:getFunds test Runs mocha tests typechain Generate Typechain typings for compiled contracts verify Verifies a contract on Etherscan or Sourcify Apart from the standarddeploy andcompile tasks, notice thetask:getFunds . We will usetask:getFunds to gettFHE tokens for deploying and interacting with the contract.

Run:

- npm
- yarn
- pnpm

npx run faucet yarn faucet pnpm faucet If you configured your .env FILE You can also use the predefined tasktask:fhenix:usefaucet by issuing the following command:

- npm
- yarn
- pnpm

npm run faucet yarn faucet pnpm faucet After a short while you should see the following message:

Done! And if you will check your accounts balance, you will notice an increase by 10 tokens. This command is useful especially if you wish to develop using Remix viaWalletConnect .

From here on now, we can proceed in a standard fashion, we can run thedeploy task to compile and deploy thecounter.sol contract. And we can even interact with it by runningtask:addCount . Or we can use Remix IDE, we just need to importFHE.sol .

import

"@fhenixprotocol/contracts/FHE.sol" ; [Edit this page]()

[Previous Setting Up]() [Next Writing the Contract]()