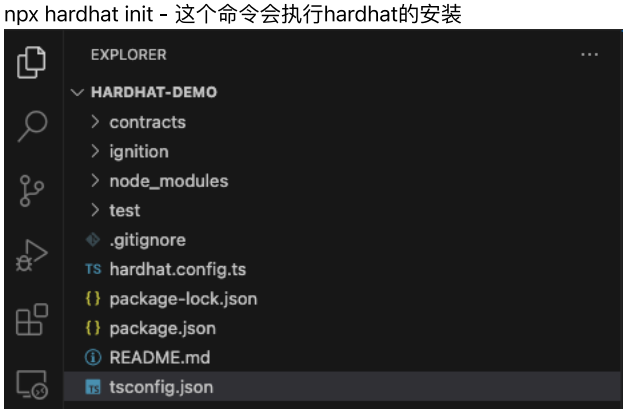
Smart contract入门分享

什么是智能合约?

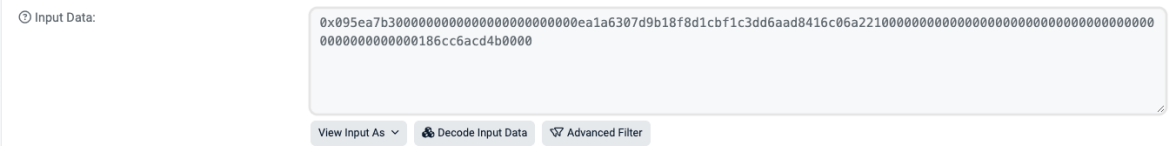
- 智能合约是运行在区块链上的程序
- 以太坊智能合约是运行在 EVM 上的智能合约
    - EVM 链：BSC, Polygon, Arbitrum(L2), Optimism(L2)

智能合约开发

- 准备工作
    - node >= 18.0
        - brew install node
    - hardhat：智能合约开发框架，包含内置的 local 以太坊网络，可以在本地完成智能合约的部署、测试、调试等工作。
        - Hardhat tutorial
- 开发
    - 使用 hardhat 创建项目
        - cd hardhat-demo && yarn init
        - npx hardhat init - 这个命令会执行hardhat的安装



        - {:height 292, :width 464}
        - npx hardhat compile
            - 编译会自动安装 solc
            - 生成 artifacts 路径，里面包含合约的 ABI 文件
                - ABI：JSON文件，定义了合约接口，包括接口名称、参数名称、参数类型、返回类型等
        - 单测 npx hardhat test
            - console.log 可以打印 solidity 变量
    - Solidity语法
        - payable - Functions and addresses declared `payable` can receive `ether` into the contract.
            - https://solidity-by-example.org/payable/
        - sending ether - transfer(不推荐), send(不推荐), **call**(推荐)
            - https://solidity-by-example.org/sending-ether/
        - calldata, memory, storage
            - storage 是状态变量，储存在区块链上，global变量
            - memory 变量是在函数方法中定义的，local变量
            - calldata 是独立的数据位置，用于存放调用的数据参数，calldata不可修改



            - {:height 94, :width 629}
            - https://etherscan.io/tx/0x8fc7c864adcf2394e905474bdcb44a8c8bf8889e2333d4775ae8a14e96fd5f63
        - Data locations 例子
        - event - 以太坊的 log，indexed 参数可以索引，一个event最多3个indexed参数
    - OpenZeppelin
        - 开源的、安全的合约库。提供了包括 Access Control, Tokens, SafeMath 等在内的合约实现
        - 使用 openzepplin 创建 ERC20 token
            - yarn add @openzeppelin/contracts@5.0.2

- pragma solidity ^0.8.24;

  import {ERC20} from "@openzeppelin/contracts/token/ERC20/ERC20.sol";

  contract FooToken is ERC20 {
      constructor() ERC20("Foo Token", "Foo") {
          uint256 initialSupply = 1000000 * (10 ** 18); // 1 million tokens
          _mint(msg.sender, initialSupply);
      }
  }
  ```

  - 如果编译失败，可能会需要安装
    - yarn add --dev "@nomicfoundation/hardhat-chai-matchers@^2.0.0" "@nomicfoundation/hardhat-ethe
      yarn add --dev "@nomicfoundation/hardhat-ignition@^0.15.0" "@nomicfoundation/ignition-core@^0

- 部署和测试
  - 添加部署脚本 scripts/deploy.ts
    - import { ethers } from "hardhat";

      async function main() {
          const fooToken = await ethers.deployContract("FooToken", []);
          await fooToken.waitForDeployment();
          console.log("FooToken deployed to:", fooToken.target);
      }
      main()
          .then(() => process.exit(0))
          .catch((error) => {
              console.error(error);
              process.exit(1);
          });
  - 本地部署
    - npx hardhat run scripts/deploy.ts --network hardhat
  - sepolia部署
    - 领水龙头 alchemy
    - 配置 hardhat.config.ts 文件，添加测试网配置 和 etherscan API key
      - networks: {
          sepolia: {
            chainId: 11155111,
            url: 'https://eth-sepolia.g.alchemy.com/v2/xxxxxxxxxxxxxx',
            accounts: [process.env.DEMO_DEPLOYER_KEY as string],
          }
        },
        etherscan: {
          apiKey: <API KEY>,
        }
      - 设置私钥 export DEMO_DEPLOYER_KEY=xxxxxxxxxxxxxxxxx
      - npx hardhat run scripts/deploy.ts --network sepolia
        - 
          
      - npx hardhat verify <address> --network sepolia
  ```

- 

  - 如何查看 etherscan

工具

- remix ethereum IDE
- sepolia faucet
- (msg.sender + nounce)
- 可升级合约
- tenderly - https://dashboard.tenderly.co/tx/mainnet/0xa8bed2bd05a2e3096ac1ec50f1b59dc6bf8ffeef2bef1f1d811e16f308067121
-