Information Technologies for Industrial Engineers

เทคโนโลยีสารสนเทศสำหรับวิศวกรอุตสาหการ

Decentralized Application (DApp)

Demo

https://nr-dapp-67.netlify.app/

What you need

- Alchemy API key
- Alchemy URL
- Addresses
  - Account address (from MetaMask)
  - Secret contract address
  - ERC20 (token) contract address
  - ERC721 (NFT) contract address
- Token Symbol

What you need

- ABIs
    - Secret contract address
    - ERC20 (token) contract address
    - ERC721 (NFT) contract address

Contract codes

- `secret.sol`
- `token.sol`
- `NFT.sol`

Setup

- `npm create vite@latest`
- …
- …
- `npm install wagmi viem@2.x @tanstack/react-query`

.env

```
VITE_ALCHEMY_KEY=
VITE_ALCHEMY_URL=
VITE_MY_ACCOUNT_ADDRESS=
VITE_SECRET_CONTRACT_ADDRESS=
VITE_TOKEN_CONTRACT_ADDRESS=
VITE_TOKEN_SYMBOL=
VITE_NFT_CONTRACT_ADDRESS=
```

Example `.env`

```
VITE_ALCHEMY_URL=
VITE_ALCHEMY_KEY=
VITE_MY_ACCOUNT_ADDRESS=0x6269f27234747F4ac12A6d5E88c75e021da290CF
VITE_SECRET_CONTRACT_ADDRESS=0xFC76182992cB0690Ca61742eB07246EB1a7705FE
VITE_TOKEN_CONTRACT_ADDRESS=0x181f336dAf6c954e7A78DA788995FbDa10B31Ef7
VITE_TOKEN_SYMBOL=POH99
VITE_NFT_CONTRACT_ADDRESS=0xe1C2DD43bE4ED63702a93b0C4b200A66F7285Cbc
```

./src/env.ts

```
export const ALCHEMY_URL = import.meta.env.VITE_ALCHEMY_URL;
export const ALCHEMY_KEY = import.meta.env.VITE_ALCHEMY_KEY;
export const MY_ACCOUNT_ADDRESS = import.meta.env.VITE_MY_ACCOUNT_ADDRESS;
export const SECRET_CONTRACT_ADDRESS = import.meta.env
  .VITE_SECRET_CONTRACT_ADDRESS;
export const TOKEN_CONTRACT_ADDRESS = import.meta.env
  .VITE_TOKEN_CONTRACT_ADDRESS;
export const NFT_CONTRACT_ADDRESS = import.meta.env.VITE_NFT_CONTRACT_ADDRESS;
export const TOKEN_SYMBOL = import.meta.env.VITE_TOKEN_SYMBOL;

if (
  !ALCHEMY_URL ||
  !ALCHEMY_KEY ||
  !MY_ACCOUNT_ADDRESS ||
  !SECRET_CONTRACT_ADDRESS ||
  !TOKEN_CONTRACT_ADDRESS ||
  !NFT_CONTRACT_ADDRESS ||
  !TOKEN_SYMBOL
) {
  throw new Error("Please set env variables correctly.");
}
```

./src/main.tsx

```tsx
import { QueryClient, QueryClientProvider } from "@tanstack/react-query";
import React from "react";
import ReactDOM from "react-dom/client";
import { createConfig, http, WagmiProvider } from "wagmi";
import { sepolia } from "wagmi/chains";
import App from "./App.tsx";
import { ALCHEMY_KEY, ALCHEMY_URL } from "./env.ts";
import "./index.css";

const queryClient = new QueryClient();

const config = createConfig({
  chains: [sepolia],
  transports: {
    [sepolia.id]: http(ALCHEMY_URL, {
      name: "Alchemy HTTP Provider",
      key: ALCHEMY_KEY,
    }),
  },
});

ReactDOM.createRoot(document.getElementById("root")!).render(
  <React.StrictMode>
    <WagmiProvider config={config}>
      <QueryClientProvider client={queryClient}>
        <App />
      </QueryClientProvider>
    </WagmiProvider>
  </React.StrictMode>
);
```

`./src/abi.ts`

```ts
export const secretABI = [
  // ...
] as const;

export const tokenABI = [
  // ...
] as const;

export const NFTABI = [
  // ...
] as const;
```

If you don't want to compile the file yourself you may use this file `abi.ts`

`./src/App.tsx`

https://gist.github.com/nnnpooh/47e64662f6667ca1d0d12f4ad359af43