DApp with Ethereum Smart Contracts - A Tutorial

Environment Setup

Here's the stuff we'll need

- NVM
- Node
- Hardhat
- MetaMask Wallet

Set up NVM first for

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
```

Add the environment variables for accesing node commands.

```
nvm install --lts
```

Install MetaMask

MetaMask will manage our wallet so that we can easily view our accounts and test our DApps easily. Go ahead and install MetaMask from `https://metamask.io/` . I'll use Firefox but you can just as well use Chrome.

(I strongly suggest you use MetaMask for this tutorial so that you can follow along. There's a million wallets out there and it would be difficult to debug issues if you use something other than MetaMask.)

Just go ahead and create an account on MetaMask, set your password and then save the recovery keys as given in the instructions.

Setting up Hardhat

In order to test out our smart contracts, we need an environment that simulates the etherium network locally. We will later try out our smart contract on a global testnet too.

First, set up a react app that we will use to interact with our environment.

```
npx create-react-app react-dapp
cd react-dapp
```

Now go ahead and set up hardhat along with all its dependencies. For now, let's not go into the details of what each part does and what alternatives are available. That would only slow you down and create confusions. It's best to get a hello world done and study options afterwards.

```
npm install ethers hardhat @nomiclabs/hardhat-waffle \
            ethereum-waffle chai \
            @nomiclabs/hardhat-ethers
```

Let's create basic hardhat configs and setup.

```
npx hardhat
```

Accept all defaults. This will create a basic sample project for us.

We need to edit the hardhat config file so that this works well with MetaMask. ChainId is a unique identifier for the blockchain.

```
vi hardhat.config.js
```

```
module.exports = {
  solidity: "0.8.4",
  paths: {                          // add this
    artifacts: './src/artifacts',   // this is where our compiled contracts will go
  },
  networks: {                       // and this ...
    hardhat: {
      chainId: 1337                 // this is needed for MetaMask
    }
  }
};
```

Start a node using:

```
npx hardhat node
```

And try to connect to it using MetaMask.