

CHEATSHEET

Setting up Hardhat

Create a project folder	<code>mkdir <name></code>
Initialise a new node	<code>npm init -y</code>
Install hardhat	<code>yarn add hardhat</code>
	<code>npm install --save-dev @nomiclabs/hardhat-waffle ethereum-waffle chai @nomiclabs/hardhat-ethers ethers</code>
Initialise new project	<code>npx hardhat</code>
Accept default settings in hh	
Open Visual Studio	<code>code .</code>
Testing	<code>npx hardhat test</code>
Setting admin permission (if nec)	<code>Set-ExecutionPolicy RemoteSigned -Scope CurrentUser</code>
OR clone from Github	<code>git clone https://github.com/yehjxraymond/hh-default.git</code>
	<code>npm i</code>
	<code>npx hardhat test</code>
	<code>code .</code>

Running Hardhat

Running scripts	<code>npx hardhat run scripts/<name>.js</code>
Access wallet	<code>npm i dotenv --save</code>
Update hardhat.config.js	<pre>require("@nomiclabs/hardhat-waffle"); require("dotenv").config(); task("accounts", "Prints the list of accounts", async () => { const accounts = await ethers.getSigners(); for (const account of accounts) { console.log(account.address); } }); module.exports = { solidity: "0.8.4", networks: { hardhat: {}, rinkeby: { url: process.env.ENDPOINT_RINKEBY, accounts: [process.env.ACCOUNT_1], }, }, };</pre>
Install OZ library	<code>npm i @openzeppelin/contracts --save</code>
Create .env	<code>ENDPOINT_RINKEBY <Alchemy>, ACCOUNT_1 <wallet private key></code>
	<code>Update .env in .gitignore</code>
Deploy on Rinkeby Etherscan	<code>npx hardhat run scripts/sample-script.js --network rinkeby</code>

NFT – Writing ERC721 Contracts

Install contract library
Create a .sol file

```
npm i @openzeppelin/contracts --save

// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.4;

import
"@openzeppelin/contracts/token/ERC721/presets/ERC721PresetMinterPau
serAutold.sol";

contract Robots is ERC721PresetMinterPauserAutold {
    constructor(
        string memory name,
        string memory symbol,
        string memory baseTokenURI
    ) ERC721PresetMinterPauserAutold(name, symbol, baseTokenURI) {}
}
```

Creating Dapp

Create React App

```
npx create-react-app garage
cd garage
yarn start
```

Set up free web hosting
Set up project repository in Github
Create new site in Netlify

Set up Tailwind CSS

```
npm install -D tailwindcss@latest postcss@latest autoprefixer@latest
<npm install -D tailwindcss@npm:@tailwindcss/postcss7-compat
postcss@^7 autoprefixer@^9>
npm install -D @craco/craco
```

Replace package.json

```
{
  "start": "craco start",
  "build": "craco build",
  "test": "craco test",
}
```

Create craco.config.js

```
// craco.config.js
module.exports = {
  style: {
    postcss: {
      plugins: [
        require('tailwindcss'),
        require('autoprefixer'),
      ],
    },
  },
}
```

Generate tailwind config

```
npx tailwindcss-cli@latest init / npx tailwindcss init
```

Replace tailwind.config.js

```
module.exports = {
  purge: [],
  darkMode: false, // or 'media' or 'class'
  theme: {
    extend: {},
  },
  variants: {
    extend: {},
  },
  plugins: [],
}
```

Replace src/index.css

```
/* ./src/index.css */
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Run development server

```
yarn start
```

Loading data from smart contracts

Install packages

```
npm i ethers axios dotenv --save
Create .env (with ALCHEMY_KEY=MY_API_KEY_GOES_HERE)
Add .env to .gitignore
```

Create config.js to read .env

```
import { config as setup } from "dotenv";
setup();
export const config = {
  alchemyKey: process.env.ALCHEMY_KEY,
};
```

Committing progress in Github

Saving progress to Github

```
git init
git add .
git commit -am "<comments on changes made>"
git push
```

Check repository

```
git remote -v
```

Change origin

```
git remote set-url origin <target>
```

First commit to github

```
git remote add origin https://github.com/markvelous/botdapp.git
git branch -M main
git push -u origin main
```