**Hướng dẫn xây dựng Token cá nhân hóa\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Biên soạn:** Nhóm Nghiên Cứu Blockchain Khoa HTTT

--------------------------------------------------------------------------------------------------------------

**TOKEN ERC20**

* Có thể tạo ra một Token ERC20
* Hiểu được Truffle-config json file
* Hiểu được quá trình deployment một dApps
* Hiểu được cách cá nhân hóa Token bằng Open-Zeppelin Smart Contracts

1. **Thư Mục Project Mẫu**

****

1. **Thư viện project**

|  |
| --- |
| npm install -g truffle |

1. **Tạo Project Truffle**

* **Bước 0:** Yêu cầu phải cài đặt các thư viện ĐÚNG tại mục 2.
* **Bước 1:** Vào đường dẫn của thư mục chứa Project: RealProject

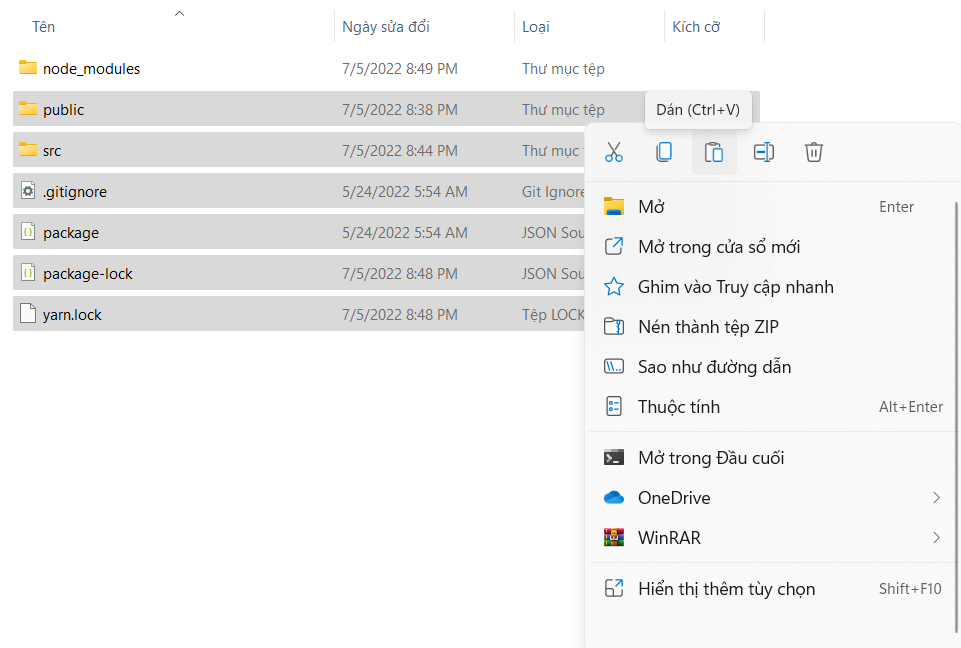


* **Bước 2:** Gõ câu lệnh **truffle unbox react**
* Sau khi unbox project truffle-react chúng ta sẽ có cấu trúc project như sau:

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Xóa thư mục client của project RealProject bởi vì phiên bản Client này đang mới cập nhật tính ổn định chưa được kiểm định. Nên sử dụng phiên bản Client cũ tại thư mục Example.
* Lưu ý chỉ sao chép những thư mục và file sau đây:



* Sau khi đưa các mục vào thư mục client tiến hành install lại thư viện bằng cách cd vào thư mục client từ thư mục root RealProject dùng lệnh **npm install.**

1. **Thêm ERC2 Tokens**

* Bước 0:
  + Vào thư mục **truffle** 🡪 **contracts** xóa contract **SimpleStorage.sol**
  + Vào thư mục **truffle 🡪 migrations** xóa tất cả các migrate có sẵn
  + Vào thư mục truffle bằng console gõ để cài đặt openzeppelin

npm install --save @openzeppelin/contracts@v3.0.0

* **Bước 1:** Tạo Contract **MyToken.sol** tại thư mục contract

Tham khảo file: MyToken.sol

* **Bước 2:** Tạo migration dùng để migrate contract: 1\_deploy\_mytoken.js có như sau:

Ảnh có chứa văn bản

Mô tả được tạo tự động

* **Bước 3:** Thiết lập môi trường chạy Project trong file truffle-config.js.

Chọn phiên bản version: “0.6.2”

Ảnh có chứa văn bản, màn hình, ảnh chụp màn hình

Mô tả được tạo tự động

* Chạy console các đoạn lệnh sau đây:

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Tiếp tục gõ console: **migrate**

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. **Thêm Crowdsale Contracts**
   * **Crowdsale:** là những đợt mở bán Token của một đồng tiền ảo nào đó trong giai đoạn ICO (Initial Coin Offering – một hình thức kêu gọi vốn đầu tư khá phổ biến trong dự án cryptocurrency)
   * **Bước 1:** Tạo Contract **Crowdsale.sol** tại thư mục contract.

Tham khảo file: **Crowdsale.sol**

* + **Bước 2:** Tạo Contract Crowdsale cho cá nhân đặt tên là **MyTokenCrowdsale.sol** tại thư mục contract.

Tham khảo file: **MyTokenCrowdsale.sol**

* + **Bước 3:** Cập nhật lại 1\_deploy\_mytoken.js như sau đây:

Tham khảo file: 1\_deploy\_mytoken.js

* + **Tiến hành migrate lại để kiểm nghiệm**

1. **Thêm KycContract**
   * **KYC** được hiểu là “know your customer” đây là điều cần thiết ứng dụng ngày nay.
   * **Cơ bản cho vấn đề này trả lời cho câu hỏi**: “Ai có thể được quyền làm việc gì?”
   * **Bước 1:** Tạo Contract **KycContract.sol** tại thư mục contract.

Tham khảo file: **KycContract.sol**

* + **Bước 2:** Cập nhật **MyTokenCrowdsale.sol**

Tham khảo file: **MyTokenCrowdsale.sol**

* + **Bước 3:** Cập nhật lại 1\_deploy\_mytoken.js như sau đây:

Tham khảo file: 1\_deploy\_mytoken.js

* + **Tiến hành migrate lại để kiểm nghiệm**

1. **React Frontend**
   * Từ bước 7 trở đi sẽ giao tiếp với frontend chính là thư mục client
   * **Tham khảo file** App.js
   * **Gõ lệnh** npm run start **để build project**
2. **Deploy Project With Metamask**

Tại thư mục truffle gõ câu lệnh truffle console --network development

Gõ tiếp **accounts**

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Vấn đề Project đang ở môi trường Ganache test. Bắt đầu từ bây giờ sẽ triển khai sang môi trường MetaMask cho người sử dụng thực tế sau này có thể sử dụng.
* Chọn một địa chỉ Ropsten Test Network trên Metamask
* Để thư chuyển một 1 ETH từ Ganache sang Metamask thử chúng ta trong console gõ như sau:

|  |
| --- |
| web3.eth.sendTransaction({from: accounts[0], to:"PASTE\_ACCOUNT\_FROM\_METAMASK", value: web3.utils.toWei("1","ether")}) |

**Ví dụ:**

web3.eth.sendTransaction({from: accounts[0], to:"0x7D6B3de67be02356fdaD4485C904e27d376A2521", value: web3.utils.toWei("1","ether")})

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. **Thêm HDWalletProvider và Mnemonic trong truffle-config**

Gõ lệnh cài đặt HDWalletProvider

|  |
| --- |
| npm install --save @truffle/hdwallet-provider@next |