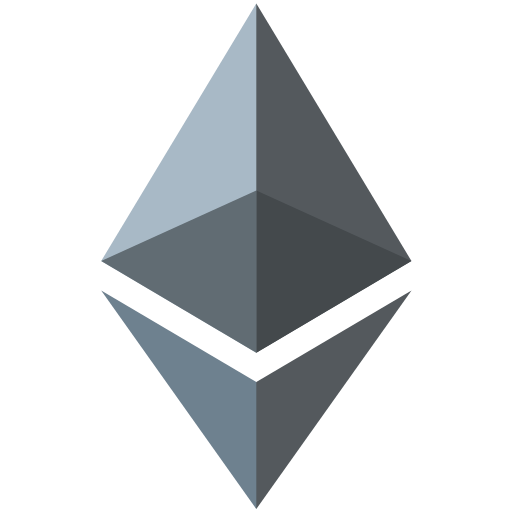


**Monnaies Numériques**



**Rendu TD5 : ERC20**

**AOUES GAYA**

1. **Git Repository**

<https://github.com/Gayardo/erc_20>

Le git a été partagé avec l’enseignant Henri Lieutaud par mail

1. **Installation de Truffle et création du projet**

On installe truffle :

**npm install -g truffle**

On crée un dossier pour le projet :

**mkdir erc\_20**

**cd erc\_20**

On initialise le projet truffle :

**truffle init**

On initialise le package npm :

**npm init -y**

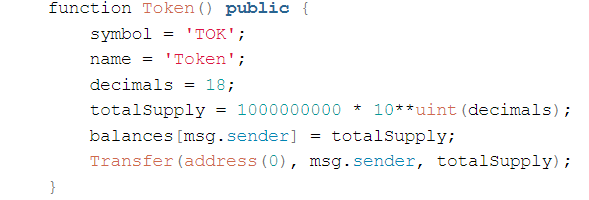
On initialise la librairie OpenZeppelin :

**npm install -E zeppelin-solidity**

1. **On crée un ERC20 Token Contract :**

Dans le dossier contracts, on crée un nouveau contrat du nom de Token.sol. On suit le modèle de erc20 sur le github de la librairie Open Zeppelin.

On crée un nom, un symbole, on choisit le nombre de décimales et le totalSupply.



1. **On implémente toutes les fonctions d’un ERC20 :**

Le code est disponible sur git dans contracts/Token.sol

Les fonctions implémentées :

**payable(), balanceOf(), transfer(),transferFrom(),approve(), allowance()**

Note : Les attributs symbol, name, decimals et totalSupply on été déclarés en public pour y avoir accès facilement.

* **On teste avec la commande truffle compile, pour voir si le contrat compile et n’a pas d’erreurs.**

1. **Migration vers Ganache :**
2. **Connexion a ganache :**

Afin d’effectuer la migration, on modifie le fichier truffle-config.js pour lui signifier la façon dont on se connecte à ganache.

On enlève les deux barres de commentaires dans la section **Développement** pour modifier les champs suivants :

Host : « 127.0.0.1 »

Port :7545

Network id : « \* »

1. **Script pour la migration :**

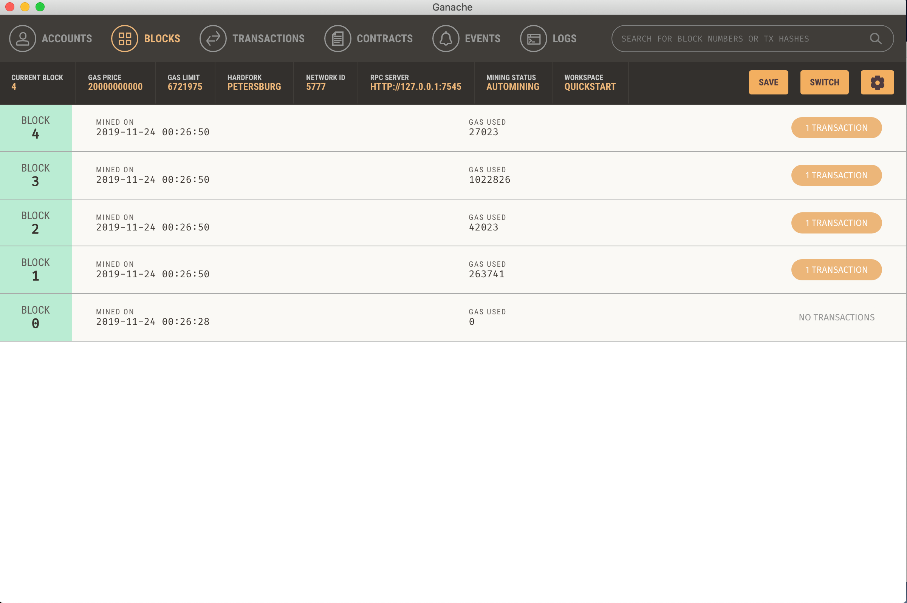
Dans le dossier migrations, on crée un fichier **2\_deploy\_contracts.js** dans lequel on met ce bout de code.



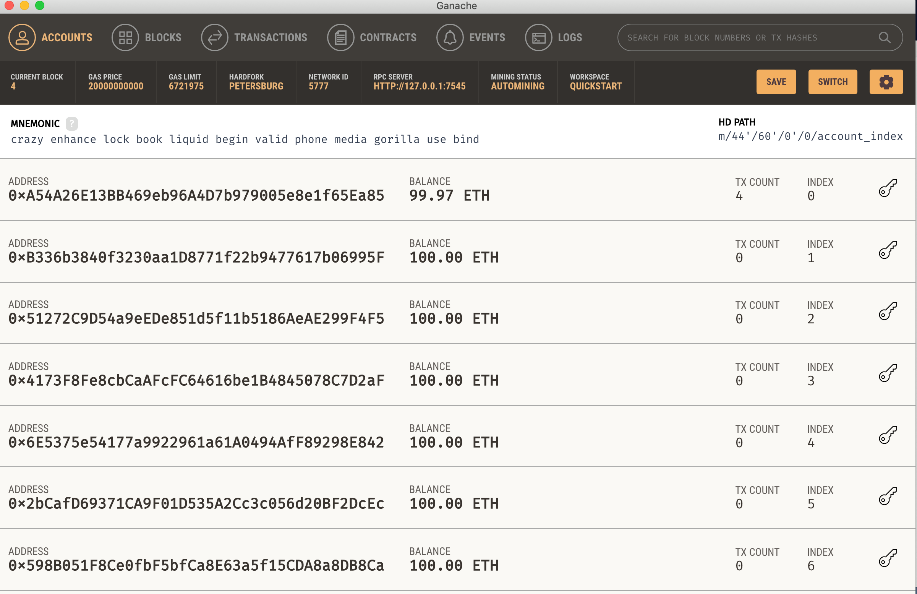
Ensuite, on lance la migration avec la commande :

**truffle migrate**

On peut alors voir les blocks qui ont été créés sur ganache :



Et la réduction des ethers du premier account.



1. **Customer WhiteListing :**

Whitelist ou liste blanche en français, il s’agit des clients qu’on considère comme sùrs et qui peuvent effectuer des transferts.

Pour l’implémenter, on utilise les contrats : **Roles.sol WhitelistAdminRole.sol** et **WhitelistedRole.sol**

On fait hériter notre contrat **Token.sol** du contrat **WhitelistedRole.sol** qui lu même hérite du contrat **WhitelistAdminRole.sol.** Enfin, les deux derniers contrats utilisent Roles.sol .

Il suffit alors de modifier les signatures des fonctions **transfer** et **transferFrom** dans notre Token.sol en ajoutant le mot **OnlyWhitelisted** pour faire en sorte que seuls les clients whitelistés puissent exécuter ces 2 fonctions.

function transfer(adress to, uint256 value)

onlyWhitelisted

public

returns (bool)

{

……….

}

{

}

Pareil pour la fonction transferFrom.