

PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua



1- truffle installation :

we execute the command : **npm install -g truffle**

```
npm install truffle
Microsoft Windows [version 10.0.22621.2428]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\LENOVO\Desktop\GL5\blockchain>npm install -g truffle
npm WARN cli npm v10.1.0 does not support Node.js v18.16.0. This version of npm supports the following node versions: '^18.17.0 || >=20.5.0'. You can find the latest version at https://nodejs.org/.

[ ] \ idealTree:npm: timing idealTree:#root Completed in 2982ms
[ ] / idealTree:npm: timing idealTree:#root Completed in 2982ms
[ ] - idealTree:npm: timing idealTree:#root Completed in 2982ms
[ ] - idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] / idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] \ idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] \ idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] | idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0
[ ] / idealTree:mocha: sill placeDep node_modules/truffle find-up@2.1.0 0

92 packages are looking for funding
  run 'npm fund' for details

C:\Users\LENOVO\Desktop\GL5\blockchain>mkdir pet-shop-tutorial
C:\Users\LENOVO\Desktop\GL5\blockchain>cd pet-shop-tutorial
C:\Users\LENOVO\Desktop\GL5\blockchain\pet-shop-tutorial>truffle
unbox pet-shop

Starting unbox...
=====
/ Preparing to download box
/ Downloading

npm WARN cli npm v10.1.0 does not support Node.js v18.16.0. This version of npm supports the following node versions: '^18.17.0 || >=20.5.0'. You can find the latest version at https://nodejs.org/.
npm WARN old lockfile
npm WARN old lockfile The package-lock.json file was created with an old version of npm,
npm WARN old lockfile so supplemental metadata must be fetched from the registry.
npm WARN old lockfile
```

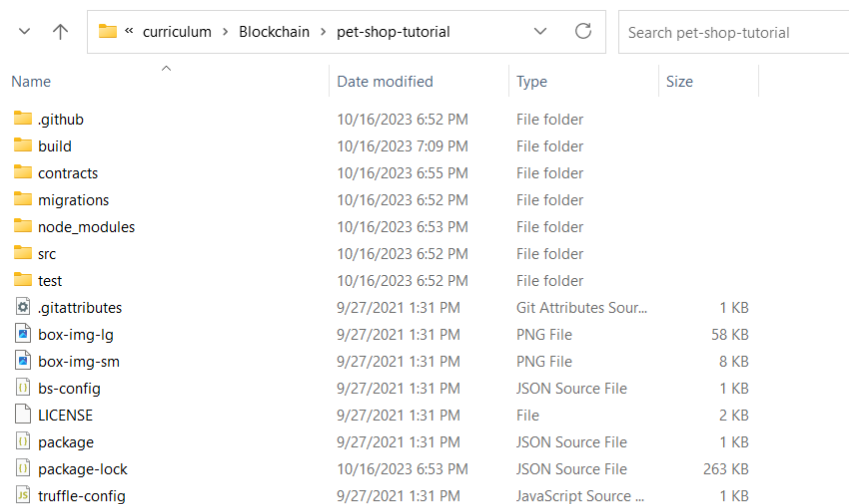
to check that truffle is installed, we used the command: **truffle version** which allows us to see the version of truffle installed

```
C:\Users\raoua>truffle version
Truffle v5.11.5 (core: 5.11.5)
Ganache v7.9.1
Solidity v0.5.16 (solc-js)
Node v16.14.0
Web3.js v1.10.0
```

2- Creating a truffle project using Truffle Box:

we start by creating the “pet-shop-tutorial” folder where we will execute our project

- To start we will use a Truffle Box dedicated to this tutorial: pet-shop . So we use **truffle unbox pet-shop** to unpack the Truffle Box we will then have this structure



Name	Date modified	Type	Size
.github	10/16/2023 6:52 PM	File folder	
build	10/16/2023 7:09 PM	File folder	
contracts	10/16/2023 6:55 PM	File folder	
migrations	10/16/2023 6:52 PM	File folder	
node_modules	10/16/2023 6:53 PM	File folder	
src	10/16/2023 6:52 PM	File folder	
test	10/16/2023 6:52 PM	File folder	
.gitattributes	9/27/2021 1:31 PM	Git Attributes Sour...	1 KB
box-img-lg	9/27/2021 1:31 PM	PNG File	58 KB
box-img-sm	9/27/2021 1:31 PM	PNG File	8 KB
bs-config	9/27/2021 1:31 PM	JSON Source File	1 KB
LICENSE	9/27/2021 1:31 PM	File	2 KB
package	9/27/2021 1:31 PM	JSON Source File	1 KB
package-lock	10/16/2023 6:53 PM	JSON Source File	263 KB
truffle-config	9/27/2021 1:31 PM	JavaScript Source ...	1 KB

3-Writing the smart contract :

What is a Smart Contract :

Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met.

—IBM—

- We begin by creating our first smart contract entitled Adoption .We indicate first the adopted variable which represents the adopters address for each pet (we have 16 pets) . Then we created the function Adopting a pet , that will allow us to associate the adopter to the adopted pet . Furthermore , we added a function getAdopters to be able to retrieve the list of adopters

PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua



```
Welcome Adoption.sol X
contracts > Adoption.sol
1  pragma solidity ^0.5.0;
2
3  contract Adoption {
4      address[16] public adopters;
5
6      // Adopting a pet
7      function adopt(uint petId) public returns (uint) {
8          require(petId >= 0 && petId <= 15);
9
10         adopters[petId] = msg.sender;
11
12         return petId;
13     }
14
15     // Retrieving the adopters
16     function getAdopters() public view returns (address[16] memory) {
17         return adopters;
18     }
19
20 }
```

4- Compiling and migrating the smart contract

4.1 Compilation :

- to compile our solidity file we use : **truffle compile**

```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS C:\Users\LENOVO\Desktop\GL5\blockchain\pet-shop-tutorial> truffle compile
>>

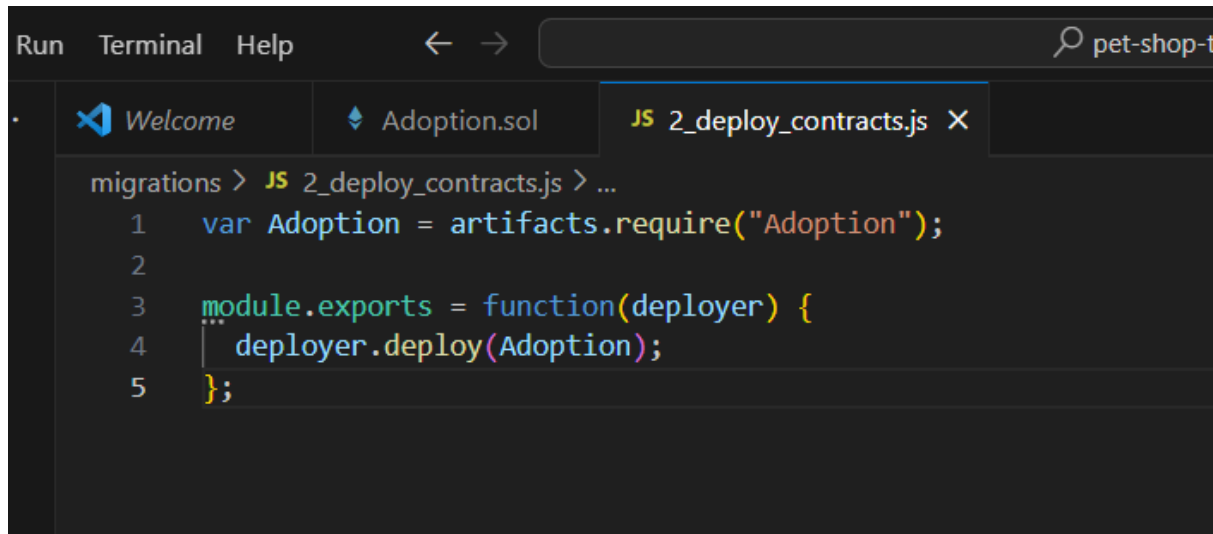
Compiling your contracts...
=====
> Compiling .\contracts\Adoption.sol
> Compiling .\contracts\Migrations.sol
> Artifacts written to C:\Users\LENOVO\Desktop\GL5\blockchain\pet-shop-tutorial\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang
PS C:\Users\LENOVO\Desktop\GL5\blockchain\pet-shop-tutorial> |
```

PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua

4.2 Migration :

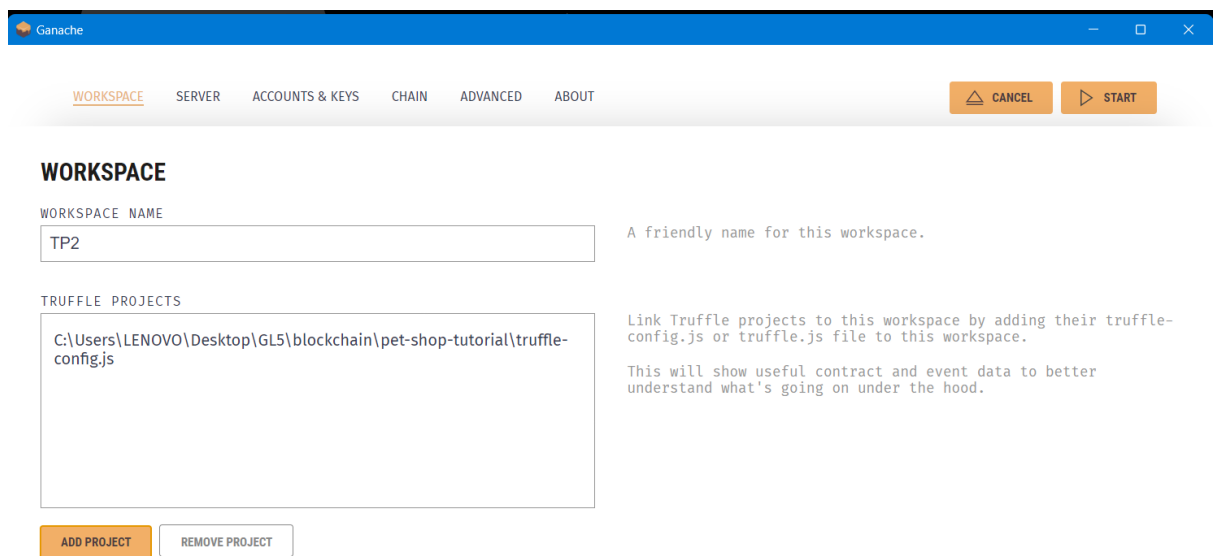
- we start by creating our own migration script :



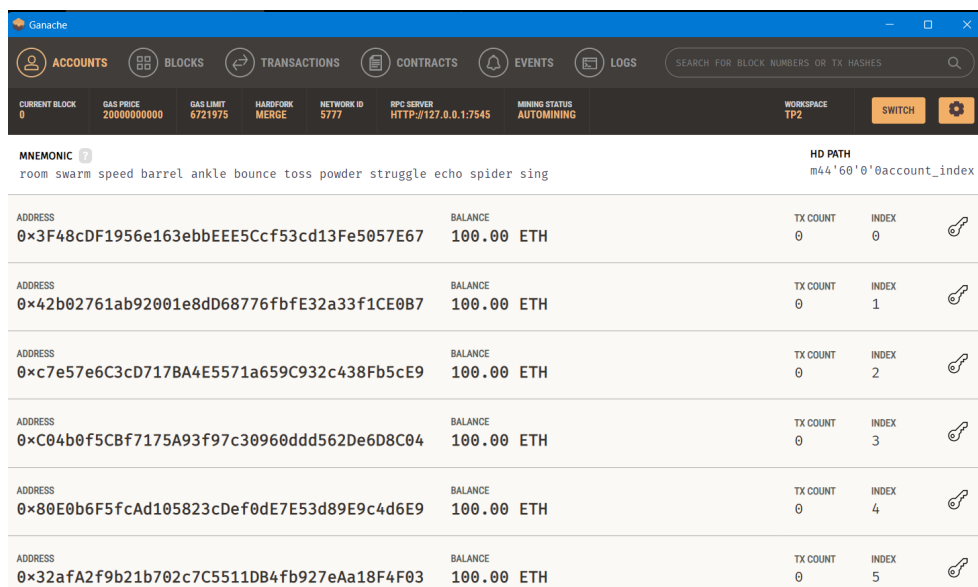
```

migrations > JS 2_deploy_contracts.js > ...
1  var Adoption = artifacts.require("Adoption");
2
3  module.exports = function(deployer) {
4    deployer.deploy(Adoption);
5  };
  
```

- to run the blockchain , we downloaded and set-up Ganache as Following : We named our workspace TP2 and added the config file to the Truffle Projects



- after clicking on start we get the following interface :



PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua

- to migrate the contract to the blockchain we execute : **truffle migrate**

```
PS C:\Users\LENOVO\Desktop\GL5\blockchain\pet-shop-tutorial> truffle migrate
>>

Compiling your contracts...
=====
> Compiling .\contracts\Adoption.sol
> Compiling .\contracts\Migrations.sol
> Artifacts written to C:\Users\LENOVO\Desktop\GL5\blockchain\pet-shop-tutorial\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang

Starting migrations...
=====
> Network name:    'development'
> Network id:     5777
> Block gas limit: 6721975 (0x6691b7)

1_initial_migration.js
=====

Deploying 'Migrations'
-----
> transaction hash: 0x5cea359be1658f58608d84260dc5dd501c242440bcf960bc615c002634b82953
> Blocks: 0        Seconds: 0
> contract address: 0x06bDe0469509DE4D1213D0283372790544C2b270
> block number:    1
> block timestamp: 1697831059
> account:         0x3F48cDF1956e163ebbEEE5CcF53cd13Fe5057E67
> balance:         99.999347804875
> gas used:        193243 (0x2f2db)
> gas price:       3.375 gwei
> value sent:      0 ETH
> total cost:      0.000652195125 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost:      0.000652195125 ETH

2_deploy_contracts.js
=====
```

```
2_deploy_contracts.js
=====

Deploying 'Adoption'
-----
> transaction hash: 0x35326d4f6d03bae58327accac994d8401b73f0922e214e1f1a0097b1ad085966
> Blocks: 0        Seconds: 0
> contract address: 0x5DbBAa1072270FA876092F456009294F986ce601
> block number:    3
> block timestamp: 1697831059
> account:         0x3F48cDF1956e163ebbEEE5CcF53cd13Fe5057E67
> balance:         99.998550649218314381
> gas used:        203827 (0x31c33)
> gas price:       3.176737487 gwei
> value sent:      0 ETH
> total cost:      0.000647504871762749 ETH

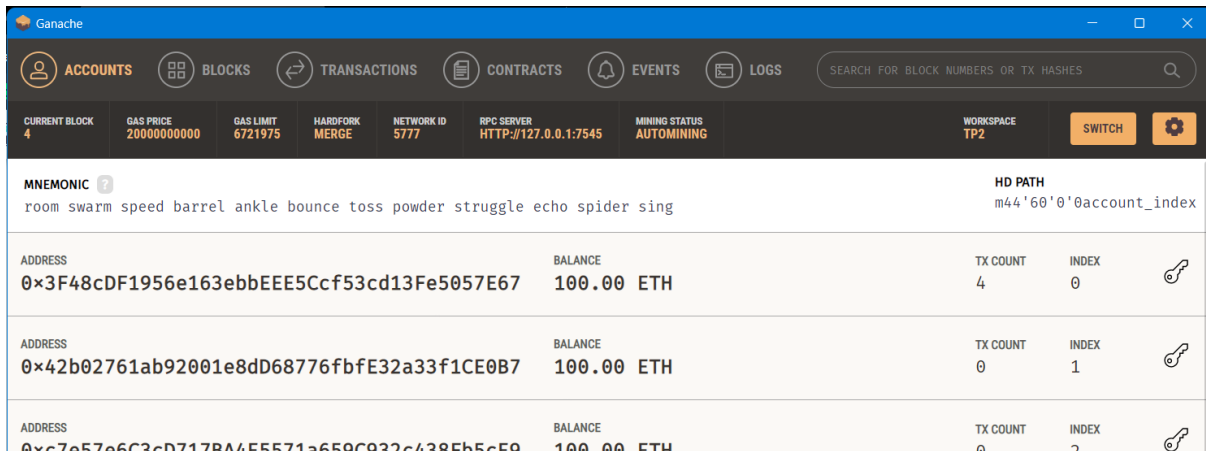
> Saving migration to chain.
> Saving artifacts
-----
> Total cost:      0.000647504871762749 ETH

Summary
=====
> Total deployments: 2
> Final cost:       0.001299699996762749 ETH
```

PetShop truffle Tutorial

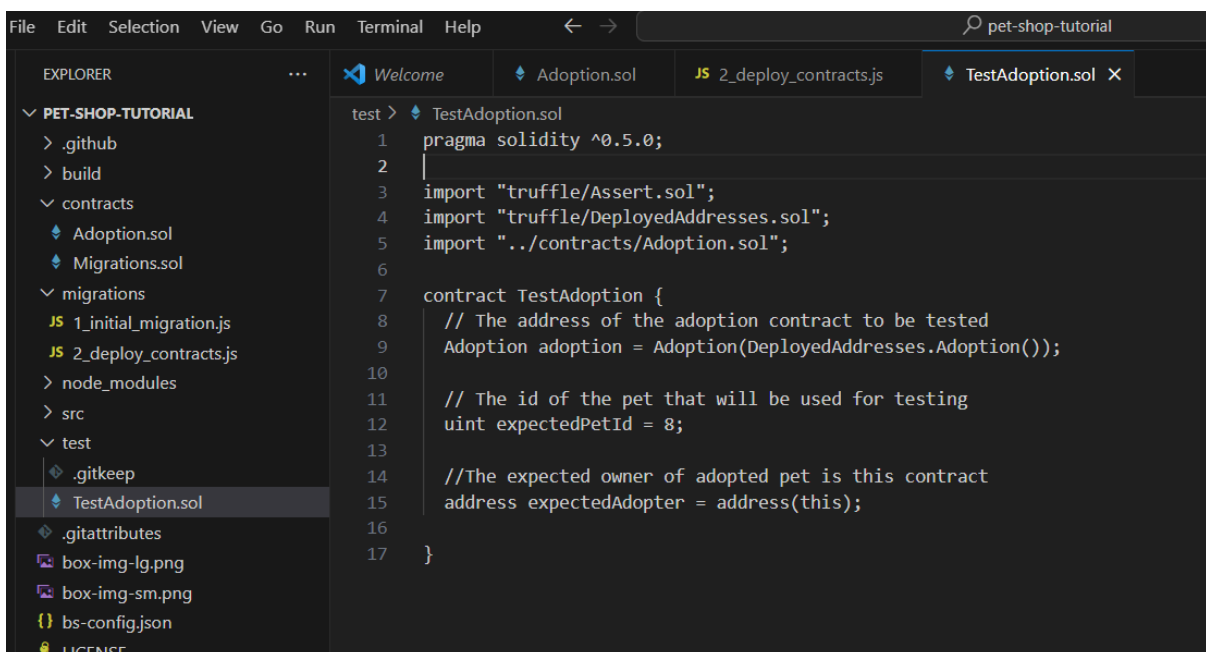
Laribi Fatma - Kahla Soulaïma - Trimech Raoua

- in the Ganache interface , we noticed that our current block number changed from 0 to 4 due to the migration of our blocks , Furthermore , we noticed that we have lower than 100 Ether



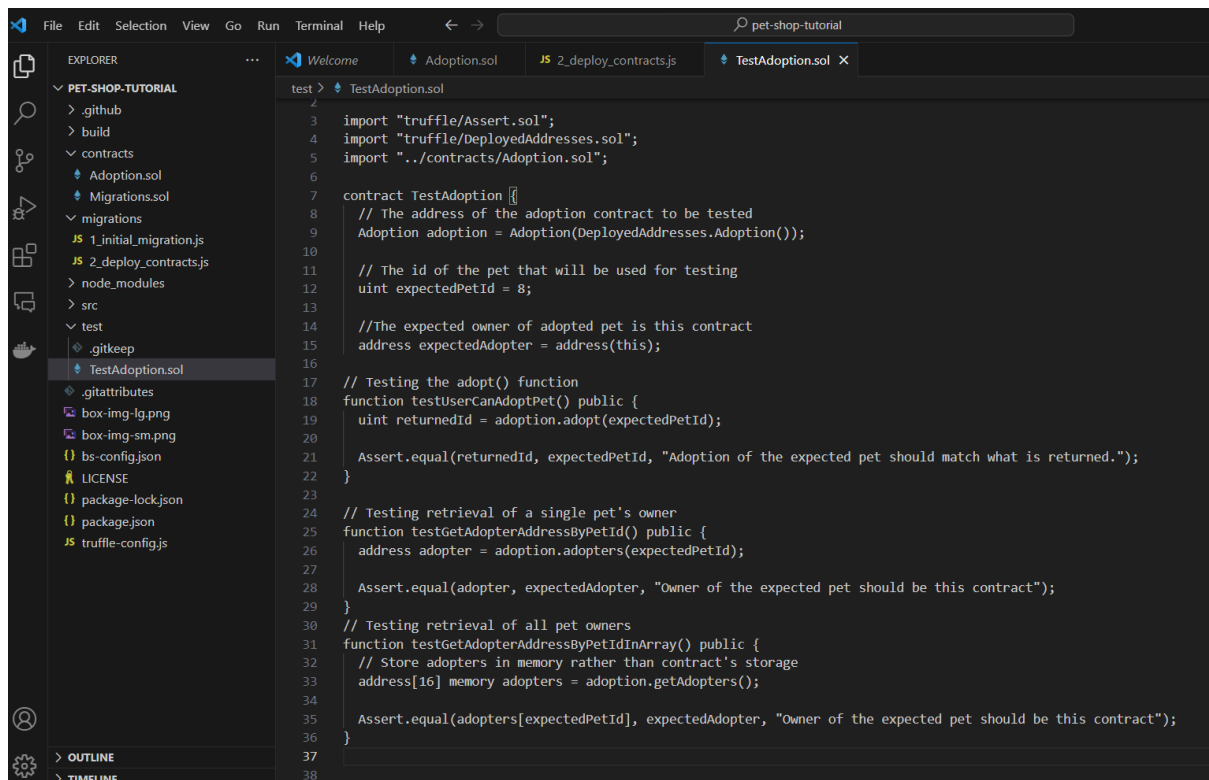
5- Testing the smart contract using Solidity :

- Truffle allows us to create tests for our smart contracts. In this case, we will be writing tests in Solidity .
 - We first started by testing the adopting function , by comparing the expected PetId to the petId returned to the function.
 - Secondly , we tested the retrieval of a single pet's owner by getting the adopter's address by the PetId and comparing it to the expected Adopter.
 - And finally we tested the retrieval of all pet owners, we first got the list of the adopters and stored it in the memory to be able to retrieve fully as one entity and then we compared the expected Adopter to the Adopter stored in this list by referring to the expectedPetId.



PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua



```

test > TestAdoption.sol
1
2
3 import "truffle/Assert.sol";
4 import "truffle/DeployedAddresses.sol";
5 import "../contracts/Adoption.sol";
6
7 contract TestAdoption {
8     // The address of the adoption contract to be tested
9     Adoption adoption = Adoption(DeployedAddresses.Adoption());
10
11     // The id of the pet that will be used for testing
12     uint expectedPetId = 8;
13
14     //The expected owner of adopted pet is this contract
15     address expectedAdopter = address(this);
16
17     // Testing the adopt() function
18     function testUserCanAdoptPet() public {
19         uint returnedId = adoption.adopt(expectedPetId);
20
21         Assert.equal(returnedId, expectedPetId, "Adoption of the expected pet should match what is returned.");
22     }
23
24     // Testing retrieval of a single pet's owner
25     function testGetAdopterAddressByPetId() public {
26         address adopter = adoption.adopters(expectedPetId);
27
28         Assert.equal(adopter, expectedAdopter, "Owner of the expected pet should be this contract");
29     }
30
31     // Testing retrieval of all pet owners
32     function testGetAdopterAddressByPetIdInArray() public {
33         // Store adopters in memory rather than contract's storage
34         address[16] memory adopters = adoption.getAdopters();
35
36         Assert.equal(adopters[expectedPetId], expectedAdopter, "Owner of the expected pet should be this contract");
37     }
38

```

6- Testing the smart contract using JavaScript :

- in this section we rewrote the previous tests that wrote in Solidity in JavaScript

PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua



```
Terminal  Help  pet-shop-tutorial

Welcome  Adoption.sol  JS 2_deploy_contracts.js  TestAdoption.sol  JS testAdoption.test.js X

test > JS testAdoption.test.js > contract("Adoption") callback > describe("adopting a pet and retrieving account addresses") callback
1  const Adoption = artifacts.require("Adoption");
2
3  contract("Adoption", (accounts) => {
4      let adoption;
5      let expectedPetId;
6
7      before(async () => {
8          adoption = await Adoption.deployed();
9      });
10
11     describe("adopting a pet and retrieving account addresses", async () => {
12         before("adopt a pet using accounts[0]", async () => {
13             await adoption.adopt(8, { from: accounts[0] });
14             expectedAdopter = accounts[0];
15         });
16
17         it("can fetch the address of an owner by pet id", async () => {
18             const adopter = await adoption.adopters(8);
19             assert.equal(adopter, expectedAdopter, "The owner of the adopted pet should be the first account.");
20         });
21
22         it("can fetch the collection of all pet owners' addresses", async () => {
23             const adopters = await adoption.getAdopters();
24             assert.equal(adopters[8], expectedAdopter, "The owner of the adopted pet should be in the collection.");
25         });
26     });
27 });
28
```

7- Running the tests:

- to run the tests we execute the command : **truffle test**
we noticed that in our Ganache interface 5 Blocks are added

PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua



PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
● PS C:\Users\LENOVO\Desktop\GL5\blockchain\pet-shop-tutorial> truffle test
>>
Using network 'development'.

Compiling your contracts...
=====
> Compiling .\contracts\Adoption.sol
> Compiling .\contracts\Migrations.sol
> Compiling .\test\TestAdoption.sol
> Compiling truffle\Assert.sol
> Compiling truffle\AssertAddress.sol
> Compiling truffle\AssertAddressArray.sol
> Compiling truffle\AssertBalance.sol
> Compiling truffle\AssertBool.sol
> Compiling truffle\AssertBytes32.sol
> Compiling truffle\AssertBytes32Array.sol
> Compiling truffle\AssertGeneral.sol
> Compiling truffle\AssertInt.sol
> Compiling truffle\AssertIntArray.sol
> Compiling truffle\AssertString.sol
> Compiling truffle\AssertUint.sol
> Compiling truffle\AssertUintArray.sol
> Compiling truffle\DeployedAddresses.sol
> Artifacts written to C:\Users\LENOVO\AppData\Local\Temp\test--10524-Qin7cgTGXzmV
> Compiled successfully using:
  - solc: 0.5.16+commit.9c3226ce.Emscripten.clang

TestAdoption
  ✓ testUserCanAdoptPet (84ms)
  ✓ testGetAdopterAddressByPetId (72ms)
  ✓ testGetAdopterAddressByPetIdInArray (95ms)

Contract: Adoption
  adopting a pet and retrieving account addresses
    ✓ can fetch the address of an owner by pet id
    ✓ can fetch the collection of all pet owners' addresses

5 passing (7s)
```

8- Creating a user interface to interact with the smart contract :

8.1 Instantiating web3 :

- we start by linking our front end to our Ethereum blockchain by using the **web3 JavaScript library**

```
initWeb3: async function() {  
  // Modern dapp browsers...  
  if (window.ethereum) {  
    App.web3Provider = window.ethereum;  
    try {  
      // Request account access  
      await window.ethereum.enable();  
    } catch (error) {  
      // User denied account access...  
      console.error("User denied account access")  
    }  
  }  
  // Legacy dapp browsers...  
  else if (window.web3) {  
    App.web3Provider = window.web3.currentProvider;  
  }  
  // If no injected web3 instance is detected, fall back to Ganache  
  else {  
    App.web3Provider = new Web3.providers.HttpProvider('http://localhost:7545');  
  }  
  web3 = new Web3(App.web3Provider);  
}
```

8.2 Instantiating the contract

- To keep migrations synchronized with the contract's information, we need to initiate our smart contract so that we don't need to change the contract's deployed address manually . We insert the following code in our app.js file

```
initContract: function() {  
  $.getJSON('Adoption.json', function(data) {  
    // Get the necessary contract artifact file and instantiate it with @truffle/contract  
    var AdoptionArtifact = data;  
    App.contracts.Adoption = TruffleContract(AdoptionArtifact);  
  
    // Set the provider for our contract  
    App.contracts.Adoption.setProvider(App.web3Provider);  
  
    // Use our contract to retrieve and mark the adopted pets  
    return App.markAdopted();  
  });  
  return App.bindEvents();  
},  
  
bindEvents: function() {  
  $(document).on('click', '.btn-adopt', App.handleAdopt);  
},  
}
```

9- Getting The Adopted Pets and Updating The UI

- With add the following code to our app.js file to be able to coordinate between the front and our blockchain so that the front end will take into consideration the already adopted pets and disable their adoption button

```
markAdopted: function() {  
  var adoptionInstance;  
  
  App.contracts.Adoption.deployed().then(function(instance) {  
    adoptionInstance = instance;  
  
    return adoptionInstance.getAdopters.call();  
  }).then(function(adopters) {  
    for (i = 0; i < adopters.length; i++) {  
      if (adopters[i] !== '0x0000000000000000000000000000000000000000') {  
        $('.panel-pet').eq(i).find('button').text('Success').attr('disabled', true);  
      }  
    }  
  }).catch(function(err) {  
    console.log(err.message);  
  });  
},  
},
```

10- Handling the adopt() Function :

- In our app.js file , we add the following code to describe the adoption process and linked it to our ethereum account so that we can execute the fee of this transaction.

PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua

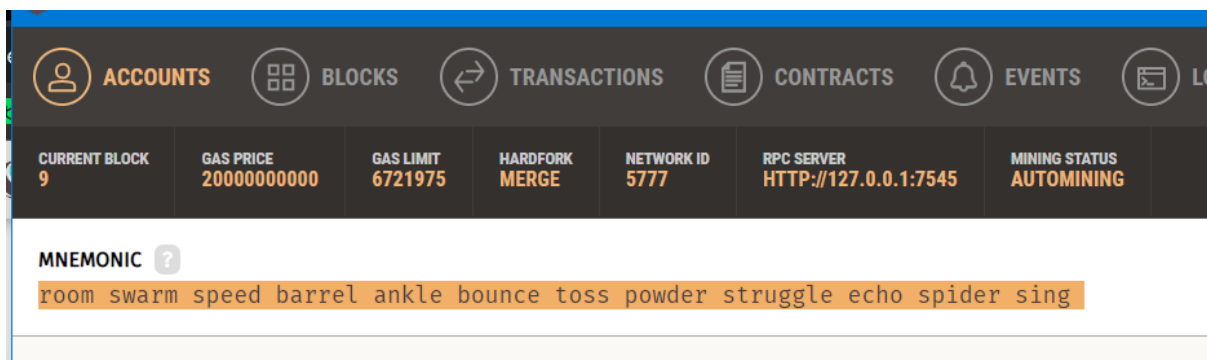


```
terminal  Help  pet-shop-tutorial
Adoption.sol  JS 2_deploy_contracts.js  TestAdoption.sol  JS testAdoption.test.js  JS app.js  X
src > js > JS app.js > handleAdopt
87
88   handleAdopt: function(event) {
89     event.preventDefault();
90
91     var petId = parseInt($(event.target).data('id'));
92
93     var adoptionInstance;
94
95     web3.eth.getAccounts(function(error, accounts) {
96       if (error) {
97         console.log(error);
98       }
99
100      var account = accounts[0];
101
102      App.contracts.Adoption.deployed().then(function(instance) {
103        adoptionInstance = instance;
104
105        // Execute adopt as a transaction by sending account
106        return adoptionInstance.adopt(petId, {from: account});
107      }).then(function(result) {
108        return App.markAdopted();
109      }).catch(function(err) {
110        console.log(err.message);
111      });
112    });
113  }
```

11- Interacting with the dapp in a browser

11.1 Installing and configuring MetaMask :

- In this section we will installing MetaMask which is a popular and widely used cryptocurrency wallet and decentralized application (DApp) browser extension. It primarily serves as a bridge between your web browser and the Ethereum blockchain, allowing users to interact with Ethereum-based applications and manage their cryptocurrency assets, including Ether (ETH) and various Ethereum-based tokens. Here are the different step we went through to set up our MetMask extension



we past the MNEMONIC in the first column of the secret phrase :

votre identité, restaurer votre portefeuille et définir un nouveau mot de passe. Veuillez tout d'abord saisir la phrase secrète de récupération qui vous a été fournie lorsque vous avez créé votre portefeuille. [En savoir plus](#)

Saisissez votre phrase secrète de récupération

J'ai une phrase de 12 mots ✓

i Vous pouvez coller toute votre phrase de récupération secrète dans n'importe quel champ

1.	2.	3.
4.	5.	6.
7.	8.	9.
10.	11.	12.

PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua



Ethereum Mainnet

Account 1

0x3F4...7E67

0 ETH

\$0.00 USD

Acheter

Envoyer

Swap

Pont

Portefe...

Jetons

NFT

Activité

Ethereum

0 ETH

\$0.00 USD

+ Importer des jetons

Rafraîchir la liste

Assistance MetaMask

Réseaux > Ajouter un réseau > Ajouter manuellement un réseau

i Un fournisseur de réseau malveillant peut mentir quant à l'état de la blockchain et enregistrer votre activité réseau. N'ajoutez que des réseaux personnalisés auxquels vous faites confiance.

Nom du réseau

New Network

Nouvelle URL de RPC

http://127.0.0.1:7545

ID de chaîne **i**

0x539

Symbole de la devise

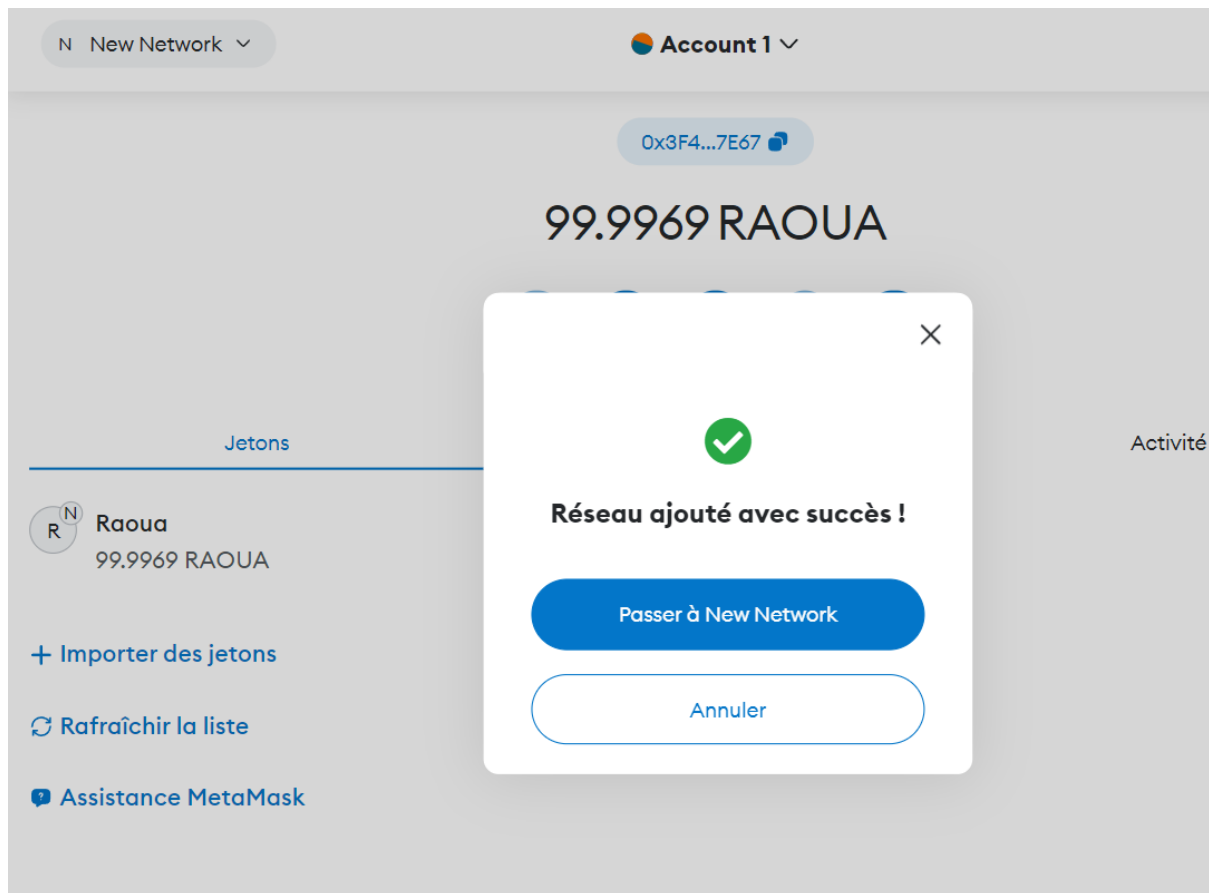
Raoua

Les données de vérification de code mnémorique sont actuellement indisponibles. Assurez-vous que le code que vous avez saisi est correct. Il aura une incidence sur les taux de conversion que vous voyez pour ce réseau

URL de l'explorateur de blocs (Facultatif)

PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua



11.2 Installing and configuring lite-server :

- Lite-server, often written as "lite-server" or "liteServer," is a simple and lightweight development server for web applications. It is primarily used for local development and testing of web applications and websites. Lite-server is often associated with front-end development, particularly when working on HTML, CSS, and JavaScript files. In our case we used it to be able to link our contracts files and the front end file under `./src` and `./build/contracts` . Furthermore , we used it to specify our dev run command .

PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua

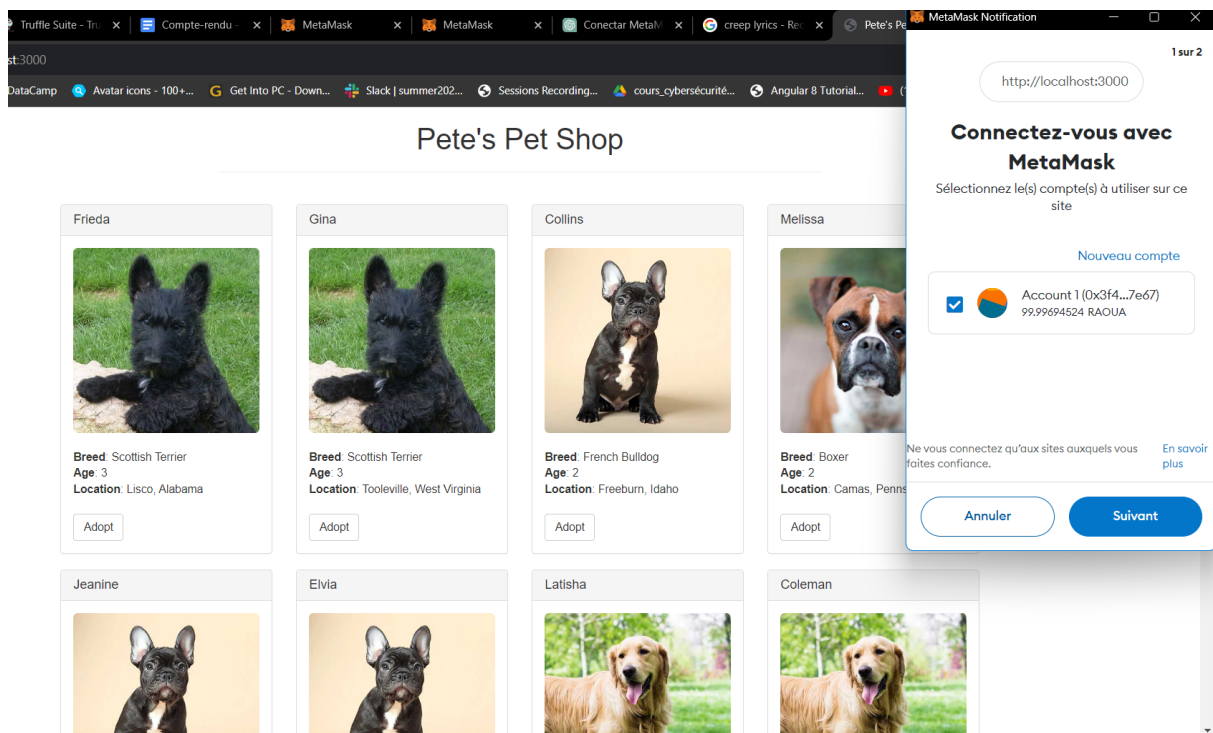
```
PS C:\Users\LENOVO\Desktop\GL5\blockchain\pet-shop-tutorial> npm run dev
npm WARN cli npm v10.1.0 does not support Node.js v18.16.0. This version of npm supports the following node versions: `^18.17.0 || >=20.5.0`. You can find the latest version at https://nodejs.org/.

> pet-shop@1.0.0 dev
> lite-server

** browser-sync config **
{
  injectChanges: false,
  files: [ '/*/*/*.{html,htm,css,js}' ],
  watchOptions: { ignored: 'node_modules' },
  server: {
    baseDir: [ './src', './build/contracts' ],
    middleware: [ [Function (anonymous)], [Function (anonymous)] ]
  }
}
[Browsersync] Access URLs:
-----
Local: http://localhost:3000
External: http://169.254.137.243:3000
-----
UI: http://localhost:3001
UI External: http://localhost:3001
-----
[Browsersync] Serving files from: ./src
[Browsersync] Serving files from: ./build/contracts
[Browsersync] Watching files...
23.10.20 22:41:16 200 GET /index.html
23.10.20 22:41:16 200 GET /css/bootstrap.min.css
23.10.20 22:41:16 200 GET /js/bootstrap.min.js
23.10.20 22:41:16 200 GET /js/web3.min.js
23.10.20 22:41:16 200 GET /js/app.js
23.10.20 22:41:16 200 GET /js/truffle-contract.js
23.10.20 22:41:17 404 GET /favicon.ico
23.10.20 22:41:17 200 GET /pets.json
23.10.20 22:41:17 404 GET /favicon.ico
23.10.20 22:41:17 200 GET /images/scottish-terrier.jpeg
23.10.20 22:41:17 200 GET /images/french-bulldog.jpeg
23.10.20 22:41:17 200 GET /images/boxer.jpeg
23.10.20 22:41:17 200 GET /images/golden-retriever.jpeg
[Browsersync] Reloading Browsers... (buffered 6 events)
23.10.20 22:41:17 304 GET /index.html
23.10.20 22:41:17 304 GET /css/bootstrap.min.css
```

11.3 Using the dapp :

- We run our local server by executing : **npm run dev** we will get the following page , we should also allow our MetaMask wallet to connect to our app which is shown down bellow :



< Retour

2 sur 2

http://localhost:3000

Connectez-vous à Account 1 (0x3f4...7e67)

Autoriser ce site à :



Consultez l'adresse, le solde du compte
et l'activité, et lancez des transactions

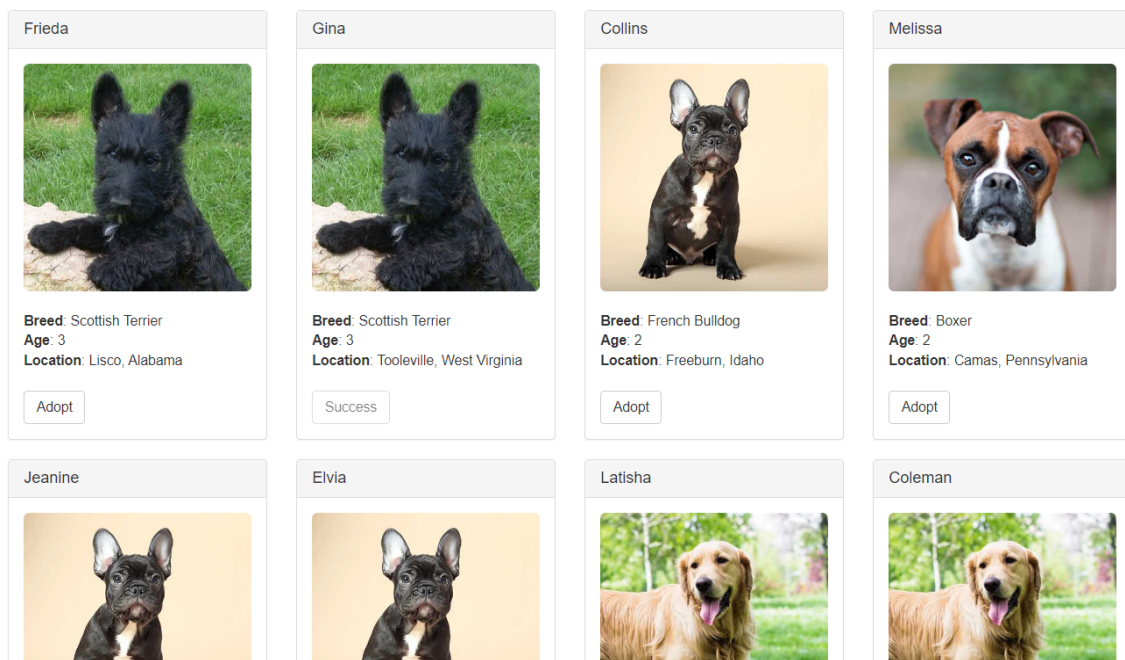
Ne vous connectez qu'aux sites auxquels vous
aitez confiance.

[En savoir
plus](#)

Annuler

Connecter

- when we click on adopt a dog we notice that in our MetaMask Wallet we are asked to confirm the transaction . Upon that we notice that our Ether count has gone down and the dog is no long available for adopting



PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua



New Network ▾

Account 1 ▾

⋮

0x3F4...7E67

99.9969 RAOUA

+

↗

↔

↶

📈

Acheter Envoyer Swap Pont Portefe...

Jetons

NFT

Activité

Oct 20, 2023

N

Adopt

Confirmé

-0 RAOUA
-0 RAOUA

Assistance MetaMask

Ganache

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK
10

GAS PRICE
20000000000

GAS LIMIT
6721975

HARDFORK
MERGE

NETWORK ID
5777

RPC SERVER
HTTP://127.0.0.1:7545

MINING STATUS
AUTOMINING

WORKSPACE
TP2

SWITCH

⚙

TX HASH

0x046e96318aafa3d3b29b7325e2aeb37f88be0ce77044a782722b4c9e03bbd243

CONTRACT CALL

FROM ADDRESS

0x3F48cDF1956e163ebbEEE5CcF53cd13Fe5057E67

TO CONTRACT ADDRESS

Adoption

GAS USED

43781

VALUE

0

TX HASH

0x142f44b115c0a4913fcb4f820fee40708f7ddd3db59d862ada7d66c6a2ba369e

CONTRACT CALL

FROM ADDRESS

0x3F48cDF1956e163ebbEEE5CcF53cd13Fe5057E67

TO CONTRACT ADDRESS

0x6A86e25133a3318eE900bcCE48A9340fEf6e4380

GAS USED

43781

VALUE

0

TX HASH

0x52223fa5a7d0a077c422496c656ef5f6f7e62facd620f8302fc1151be0012e3a

CONTRACT CALL

FROM ADDRESS

0x3F48cDF1956e163ebbEEE5CcF53cd13Fe5057E67

TO CONTRACT ADDRESS

0x7f6288e72F80ac54D8134268700ab5dE1D8f1CA9

GAS USED

28638

VALUE

0

TX HASH

0x017eab17ef5e9bf30232fa6411924f725d662036b69e194900ad75d82af6e534

CONTRACT CREATION

FROM ADDRESS

0x3F48cDF1956e163ebbEEE5CcF53cd13Fe5057E67

CREATED CONTRACT ADDRESS

0x6A86e25133a3318eE900bcCE48A9340fEf6e4380

GAS USED

203827

VALUE

0

TX HASH

0x47b85a4488cd124a73269a46b75d9727183e080cd5d325a22fca5abc8f27d50a


CONTRACT CALL

PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua

Angular 8 Tutorial...

Melissa



Breed: Boxer
Age: 2
Location: Camas, Penns

Adopt

MetaMask Notification

New Network

Account 1 → 0x5Db...e601

0x5Db...e601 : ADOPT

DÉTAILS DONNÉES HEX

Marché >

0.0006732

Carburant (estimé) 0.0006732 RAOUA

Probablement dans < 30 secondes

Frais maximaux: 0.00087434 RAOUA

0.0006732

Total 0.0006732 RAOUA

Montant + frais de carburant

Montant maximal: 0.00087434 RAOUA


Rejeter Confirmer

PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua

Pete's Pet Shop


Frieda



Breed: Scottish Terrier
Age: 3
Location: Lisco, Alabama

Adopt


Gina



Breed: Scottish Terrier
Age: 3
Location: Tooleville, West Virginia

Success


Collins



Breed: French Bulldog
Age: 2
Location: Freeburn, Idaho

Adopt

Melissa



Breed: Boxer
Age: 2
Location: Camas, Pennsylvania

Success

PetShop truffle Tutorial

Laribi Fatma - Kahla Soulaïma - Trimech Raoua



N

Account 1

0x3F4...7E67

99.9968 RAOUA

Acheter

Envoyer

Swap

Pont

Portefeu...

Jetons

NFT

Activité

Oct 20, 2023

N

Adopt

Confirmé

-0 RAOUA
-0 RAOUA

N

Adopt

Confirmé

-0 RAOUA
-0 RAOUA

Assistance MetaMask