

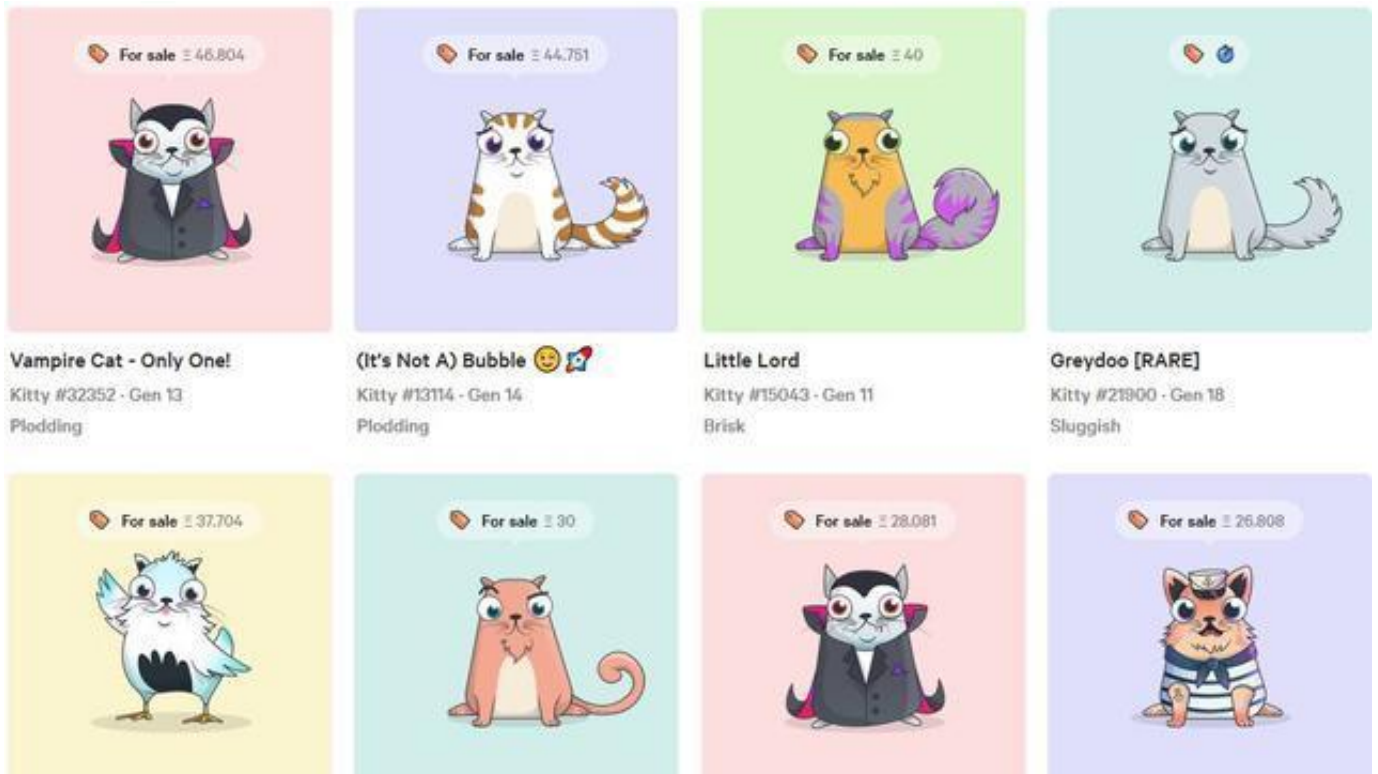
开发和部署以太坊DApp（宠物商店）

1、背景介绍以及相关问题

- 以太猫游戏介绍

CryptoKitties（以太猫）是2017年11月上线的一款以太坊区块链虚拟养猫游戏，用户可以花费以太币买卖并繁殖不同品种的虚拟宠物猫，一只虚拟宠物猫在市场最高标价为340万美元。上线不到10天一跃成为以太坊上交易量最高的DApp，12月还出现了严重的拥堵事件，也因此暴露了区块链存在的问题。

以太猫网址：<https://www.cryptokitties.co/>



- 以太坊性能优化介绍

为了改善以太坊的效率、吞吐率和并发性等问题，出现了三种以太坊性能优化的代表性技术，分别是雷电网络，分片技术和Casper共识机制。

- 雷电网络（Raiden Network）：侧链，基于以太坊的链下交易方案；将小额交易转移到链下，达到一定量或者时间后再关闭状态通道并进行结算，从而解决以太坊中转账交易的速度、费用和隐私问题。
- 分片技术（Sharding）：思路是每个节点只处理一部分交易，比如一部分账户发起的交易，从而减轻节点的计算和存储负担。
- Casper共识机制：由于PoW（工作量证明）消耗大量算力和电力，以太坊基金会一致积极推进使用PoS（权益证明）代替PoW，以太坊官方将它的PoS共识机制成为Casper。

- 开发宠物商店DApp背景

以太坊游戏很火，可以通过开发类似功能的宠物商店DApp，将学会以下内容：

- 搭建开发环境（使用到Truffle框架）
- 编写和部署智能合约到区块链
- Web3和智能合约的交互
- MetaMask的使用

项目背景：Peter有一个宠物店，店里有16只宠物等待领养，他希望用以太坊技术来开发一款去中心化应用DApp，让大家来领养宠物。

分。

在本项目中已经提供了网站结构和样式，我们只需要编写智能合约和前端逻辑。

前提条件：需要具备基本的以太坊和智能合约相关基础知识，以及具备HTML和JavaScript的基本使用

适用对象：DApp开发新手

宠物商店pet-shop官网参考地址（英文）：

<http://truffleframework.com/tutorials/pet-shop>

有些步骤可能跟官网不同，MetaMask相关的步骤省略了比较多，可以查看官网的操作。

2、搭建开发环境

- 安装Node.js

官网：<https://nodejs.org/en/>

安装很简单，只需要下载安装包直接安装即可，可以先通过终端检查安装情况再安装，没有对应结果显示再安装：

```
wenzildeiMac:~ wenzil$ npm -v
3.10.10
wenzildeiMac:~ wenzil$ node -v
v6.9.5
```

- 安装Truffle：

```
npm install -g truffle
```

3、通过Truffle Box创建项目

建一个项目的目录，然后进入到该目录，如

```
mkdir pet-shop-tutorial
cd pet-shop-tutorial
```

创建一个Truffle项目

```
wenzildeiMac:pet-shop-tutorial wenzil$ truffle unbox pet-shop
Downloading...
Unpacking...
Setting up...
Unbox successful. Sweet!
```

Commands:

```
Compile:      truffle compile
Migrate:      truffle migrate
Test contracts: truffle test
Run dev server: npm run dev
```

4、项目目录结构

- contracts/: 放智能合约Solidity代码的文件夹
- migrations/: 部署智能合约的脚本
- tests/: 存放用于测试的智能合约文件
- truffle.js: Truffle默认的配置文件

5、编写智能合约

在contracts目录下，创建一个名为Adoption.sol的合约文件。

```
pragma solidity ^ 0.4 .17;

contract Adoption {
    // address类型指向的是以太坊地址，存储为20个字节的值
    // 定义了一个固定长度16为的数组，也就是16个领养宠物的人对应的以太坊地址
    address[16] public adopters;

    // 领养宠物
    function adopt(uint petId) public returns(uint) {
        // require函数用来检查petId，确保petId在0~15之间，防止数组下标越界
        require(petId >= 0 && petId <= 15);
        // msg.sender表示调用此函数或者智能合约的地址
        adopters[petId] = msg.sender;
        // 返回petId作为确认
    }
}
```

```

    return petId;
}

// 获取领养人
// 确保返回类型是adopters指定的类型->address[16]
function getAdopters() public view returns(address[16]) {
    return adopters;
}
}

```

配置以太坊客户端本地环境：

打开truffle.js配置文件，修改端口为9545。

```

1  module.exports = {
2    // See <http://truffleframework.com/docs/advanced/configuration>
3    // for more about customizing your Truffle configuration!
4    networks: {
5      development: {
6        host: "127.0.0.1",
7        port: 9545,
8        network_id: "*" // Match any network id
9      }
10   }
11 };
12

```

6、编译和部署智能合约

Truffle集成了一个叫Truffle Develop的开发者控制台，可以用来部署和测试智能合约。

- 编译智能合约

Solidity是一种编译型语言，意味着我们需要将我的Solidity编译成以太坊虚拟机（EVM）执行的字节码。

打开终端，确保在包含DApp目录下执行如下命令：

```
truffle develop
```

```
wenzildeiMac:pet-shop-tutorial wenzil$ truffle develop
Truffle Develop started at http://127.0.0.1:9545/
```

```
Accounts:
```

```
(0) 0x627306090abab3a6e1400e9345bc60c78a8bef57
```

```
(1) 0xf17f52151ebef6c7334fad080c5704d77216b732
(2) 0xc5fdf4076b8f3a5357c5e395ab970b5b54098fef
(3) 0x821aea9a577a9b44299b9c15c88cf3087f3b5544
(4) 0x0d1d4e623d10f9fba5db95830f7d3839406c6af2
(5) 0x2932b7a2355d6fecc4b5c0b6bd44cc31df247a2e
(6) 0x2191ef87e392377ec08e7c08eb105ef5448eced5
(7) 0x0f4f2ac550a1b4e2280d04c21cea7ebd822934b5
(8) 0x6330a553fc93768f612722bb8c2ec78ac90b3bbc
(9) 0x5aeda56215b167893e80b4fe645ba6d5bab767de
```

Private Keys:

```
(0) c87509a1c067bbde78beb793e6fa76530b6382a4c0241e5e4a9ec0a0f44dc0d3
(1) ae6ae8e5ccbf04590405997ee2d52d2b330726137b875053c36d94e974d162f
(2) 0dbbe8e4ae425a6d2687f1a7e3ba17bc98c673636790f1b8ad91193c05875ef1
(3) c88b703fb08cbea894b6aeff5a544fb92e78a18e19814cd85da83b71f772aa6c
(4) 388c684f0ba1ef5017716adb5d21a053ea8e90277d0868337519f97bede61418
(5) 659cbb0e2411a44db63778987b1e22153c086a95eb6b18bdf89de078917abc63
(6) 82d052c865f5763aad42add438569276c00d3d88a2d062d36b2bae914d58b8c8
(7) aa3680d5d48a8283413f7a108367c7299ca73f553735860a87b08f39395618b7
(8) 0f62d96d6675f32685bbdb8ac13cda7c23436f63efbb9d07700d8669ff12b7c4
(9) 8d5366123cb560bb606379f90a0bfd4769eccc0557f1b362dcae9012b548b1e5
```

Mnemonic: candy maple cake sugar pudding cream honey rich smooth crumble
sweet treat

⚠ Important ⚠ : This mnemonic was created for you by Truffle. It is not secure.

Ensure you do not use it on production blockchains, or else you risk losing funds.

```
truffle(develop)>
```

然后输入"compile"进行编译:

```
truffle(develop)> compile
Compiling ./contracts/Adoption.sol...
Compiling ./contracts/Migrations.sol...
```

Compilation warnings encountered:

```
/Users/wenzil/Desktop/study/pet-shop-tutorial/contracts/Migrations.sol:1
1:3: Warning: Defining constructors as functions with the same name as the contract is deprecated. Use "constructor(...) { ... }" instead.
```

```
function Migrations() public {
```

```
^ (Relevant source part starts here and spans across multiple lines).
```

```
Writing artifacts to ./build/contracts
```

出现了警告，可以先不用管，也可以按照我另外一篇《以太坊开发简介（下）》涉及到的来修改。

- 编写部署脚本

刚才已经成功编译了智能合约，然后可以部署到区块链了。

在migrations文件夹下已经有一个1_initial_migration.js部署脚本，用来部署Migrations.sol智能合约。

然后新建一个部署脚本，比如文件名为："2_deploy_contracts.js"，格式为"2_文件名.js"

```
var Adoption = artifacts.require("Adoption");

module.exports = function(deployer) {
  deployer.deploy(Adoption);
};
```

部署智能合约之前，确保有一个区块链在运行，可以使用Ganache来开启一个私有链来进行开发和部署智能合约。运行测试。

7、编译和部署智能合约

执行"migrate"命令部署智能合约：

```
truffle(develop)> migrate
Using network 'develop'.

Running migration: 1_initial_migration.js
  Deploying Migrations...
  ... 0x76c32a791031a1c9995cd0721c514f3f51ad9dca09945962ff0674c9f99eb2cd
  Migrations: 0x8cdaf0cd259887258bc13a92c0a6da92698644c0
Saving successful migration to network...
  ... 0xd7bc86d31bee32fa3988f1c1eabce403a1b5d570340a3a9cdba53a472ee8c956
Saving artifacts...
Running migration: 2_deploy_contracts.js
  Deploying Adoption...
  ... 0x40b2148c4afb07fb51f6f3d7db3b960e33685e40e6e16e64175912da3f590352
  Adoption: 0x345ca3e014aaf5dca488057592ee47305d9b3e10
Saving successful migration to network...
  ... 0xf36163615f41ef7ed8f4a8f192149a0bf633fe1a2398ce001bf44c43dc7bdda0
Saving artifacts...
```

8、测试智能合约

在test目录下新建一个TestAdoption.sol，编写测试合约

```
pragma solidity ^ 0.4 .11;

import "truffle/Assert.sol";
import "truffle/DeployedAddresses.sol";
import "../contracts/Adoption.sol";

contract TestAdoption {
    Adoption adoption = Adoption(DeployedAddresses.Adoption());

    // 测试领养方法
    function testUserCanAdoptPet() {
        uint returnedId = adoption.adopt(8);
        uint expected = 8;
        Assert.equal(returnedId, expected, "Adoption of pet ID 8 should be recorded.");
    }

    // 测试根据给定宠物id获取领养人的函数
    function testGetAdopterAddressByPetId() public {
        // 期望领养者的地址就是本合约地址，因为交易是由测试合约发起交易，
        address expected = this;
        address adopter = adoption.adopters(8);
        Assert.equal(adopter, expected, "Owner of pet ID 8 should be recorded.");
    }

    // 测试获取所有领养人
    function testGetAdopterAddressByPetIdInArray() public {
        // 领养者的地址就是本合约地址
        address expected = this;
        address[16] memory adopters = adoption.getAdopters();
        Assert.equal(adopters[8], expected, "Owner of pet ID 8 should be recorded.");
    }
}
```

然后输入"test"进行测试：

```
truffle(develop)> test
Using network 'develop'.

Compiling ./contracts/Adoption.sol...
Compiling ./test/TestAdoption.sol...
```

```
Compiling truffle/Assert.sol...
Compiling truffle/DeployedAddresses.sol...
```

Compilation warnings encountered:

```
/Users/wenzil/Desktop/study/pet-shop-tutorial/test/TestAdoption.sol:11:3: Warning: No visibility specified. Defaulting to "public".
function testUserCanAdoptPet() {
^ (Relevant source part starts here and spans across multiple lines).
/Users/wenzil/Desktop/study/pet-shop-tutorial/contracts/Adoption.sol:20:3: Warning: Function state mutability can be restricted to view
function getAdopters() public returns(address[16]) {
^ (Relevant source part starts here and spans across multiple lines).
```

TestAdoption

- ✓ testUserCanAdoptPet (113ms)
- ✓ testGetAdopterAddressByPetId (207ms)
- ✓ testGetAdopterAddressByPetIdInArray (141ms)

3 passing (1s)

9、创建与智能合约交互的UI

现在，我们已经创建了智能合约，将其部署到我们的本地测试区块链中，并确认我们可以通过控制台与它进行交互，现在是时候创建一个UI，让Peter有一些东西可以用于他的宠物店！

这个应用程序的前端代码在pet-shop项目目录里。存在于src/目录中。打开"src/js/app.js"修改initWeb3()

修改app.js的initWeb3(), 修改为

```
initWeb3: function() {
  // Is there an injected web3 instance?
  if (typeof web3 !== 'undefined') {
    App.web3Provider = web3.currentProvider;
  } else {
    // If no injected web3 instance is detected, fall back to Ganache
    // 优先调用MetaMask提供的web3的实例
    App.web3Provider = new Web3.providers.HttpProvider('http://localhost:8545');
  }
  web3 = new Web3(App.web3Provider);
```



```
    return App.initContract();
  },
```

修改initContract()代码，如下

```
initContract: function() {
  // 加载Adoption.json, 保存了Adoption的ABI (接口说明) 信息及部署后的网络(地址)信息
  // 它在编译合约的时候生成ABI, 在部署的时候追加网络信息
  $.getJSON('Adoption.json', function(data) {
    // Get the necessary contract artifact file and instantiate it with truffle-contract
    // 用Adoption.json数据创建一个可交互的TruffleContract合约实例。
    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();
},
```

修改markAdopted()代码:

```
markAdopted: function(adopters, account) {
  var adoptionInstance;

  App.contracts.Adoption.deployed().then(function(instance) {
    adoptionInstance = instance;

    // 调用合约的getAdopters(), 用call读取信息不用消耗gas
    return adoptionInstance.getAdopters.call();
  }).then(function(adopters) {
    for (i = 0; i < adopters.length; i++) {
      if (adopters[i] !== '0x0000000000000000000000000000000000000000') {
        $('<div>'.panel-pet').eq(i).find('button').text('Success').attr('disabled', true);
      }
    }
  }).catch(function(err) {
    console.log(err.message);
  });
}
```

```
}
```

修改handleAdopt()代码：

```
handleAdopt: function(event) {
    event.preventDefault();

    var petId = parseInt($(event.target).data('id'));

    var adoptionInstance;

    // 获取用户账号
    web3.eth.getAccounts(function(error, accounts) {
        if (error) {
            console.log(error);
        }

        var account = accounts[0];

        App.contracts.Adoption.deployed().then(function(instance) {
            adoptionInstance = instance;

            // 发送交易领养宠物
            return adoptionInstance.adopt(petId, {from: account});
        }).then(function(result) {
            return App.markAdopted();
        }).catch(function(err) {
            console.log(err.message);
        });
    });
}
```

10、启动lite-server

新建一个终端，进入到pet-shop-tutorial根目录，执行如下命令：

```
npm run dev
```

运行结果显示如下：

```
wenzildeiMac:pet-shop-tutorial wenzil$ npm run dev
```

```
> pet-shop@1.0.0 dev /Users/wenzil/Desktop/study/pet-shop-tutorial
> lite-server
```

```

** browser-sync config **
{ injectChanges: false,
  files: [ './**/*.html,htm,css,js' ],
  watchOptions: { ignored: 'node_modules' },
  server:
    { baseDir: [ './src', './build/contracts' ],
      middleware: [ [Function], [Function] ] } }
[Browsersync] Access URLs:
-----
    Local: http://localhost:3000
  External: http://192.168.1.100:3000
-----
    UI: http://localhost:3001
  UI External: http://192.168.1.100:3001
-----
[Browsersync] Serving files from: ./src
[Browsersync] Serving files from: ./build/contracts
[Browsersync] Watching files...
18.05.21 00:32:36 200 GET /index.html
18.05.21 00:32:37 200 GET /css/bootstrap.min.css
18.05.21 00:32:37 200 GET /js/bootstrap.min.js
18.05.21 00:32:37 200 GET /js/web3.min.js
18.05.21 00:32:37 200 GET /js/app.js
18.05.21 00:32:37 200 GET /js/truffle-contract.js
18.05.21 00:32:39 404 GET /favicon.ico

```

11、MetaMask的使用

选择Custom RPC，添加IP地址("<http://127.0.0.1:9545/>")

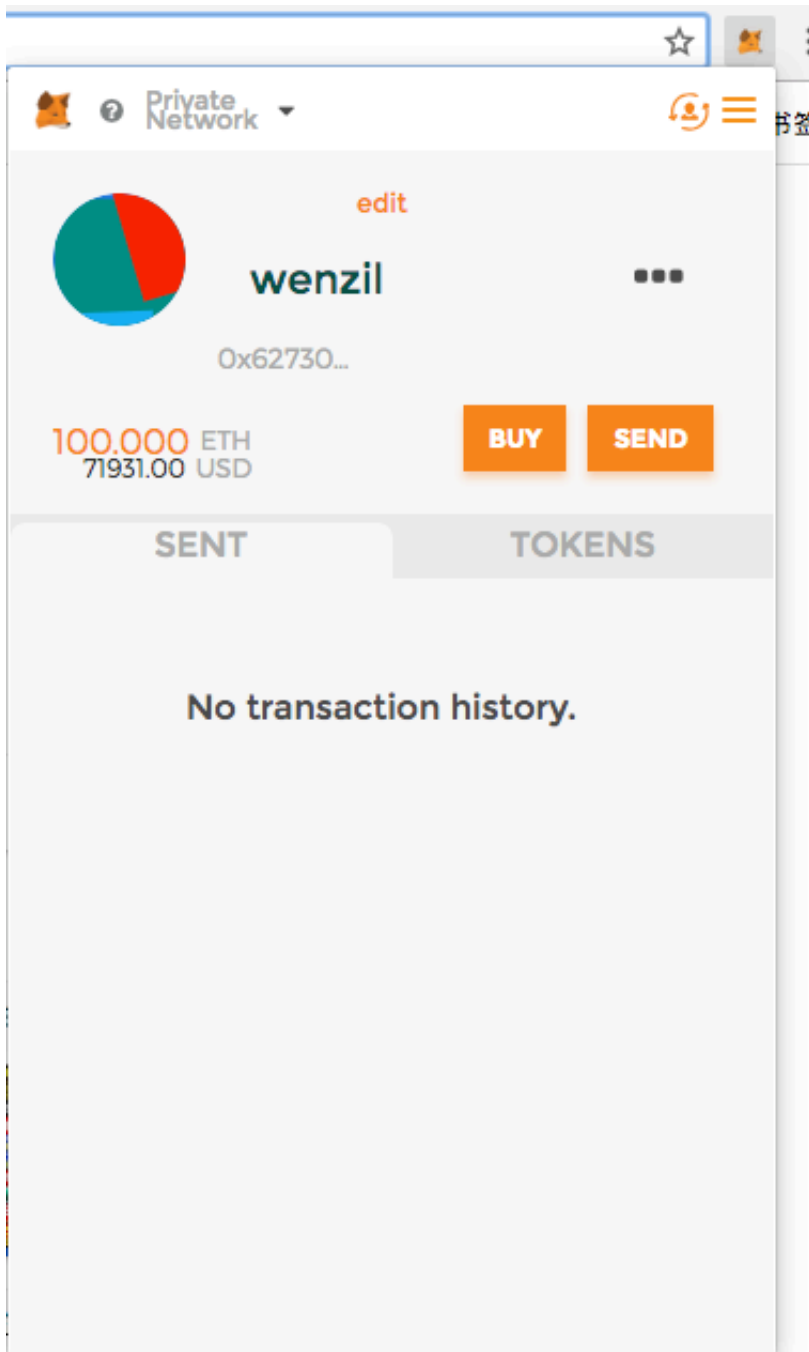
作为客户自定义RPC网络，默认帐号为以太币数量为0，可以回去查看"truffle develop"生成的帐号的私钥，如：

```

Private Keys:
(0) c87509a1c067bbde78beb793e6fa76530b6382a4c0241e5e4a9ec0a0f44dc0d3

```

然后选择"Import Account"导入私钥，获取对应的帐号，发现为100个以太币。



12、启动服务，前端测试领养宠物：

修改"src/index.html"里面jQuery的地址，因为谷歌被墙，jQuery不可用，可以换为本地的或者其他网站的jQuery地址。

如图：

```
51 </div>
52
53 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
54 <!-- <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script> -->
55 <!-- Include all compiled plugins (below), or include individual files as needed -->
56 <script src="http://code.jquery.com/jquery-latest.js"></script>
57 <script src="js/bootstrap.min.js"></script>
58 <script src="js/web3.min.js"></script>
59 <script src="js/truffle-contract.js"></script>
60 <script src="js/app.js"></script>
61 </body>
62 </html>
```


然后打开"<http://localhost:3000/>"
然后刷新页面，发现原来的空白页面多了很多宠物狗，如图：

localhost:3000

☆

Pete's Pet Shop

Frieda




Breed: Scottish Terrier

Age: 3

Location: Lisco, Alabama

Adopt

Gina




Breed: Scottish Terrier

Age: 3

Location: Tooleville, West Virginia

Adopt

Collins




Breed: French Bulldog

Age: 2

Location: Freeburn, Idaho

Adopt

Melissa




Breed: Boxer

Age: 2


Location: Camas, Pennsylvania

Adopt


Jeanine




Elvia



Latisha



Coleman



然后，点击某个宠物的"Adopt"按钮，会弹出交易确认弹框，点击"SUBMIT"即可。

MetaMask Notification

CONFIRM TRANSACTION Private Network

wenzli
627306...Ef57
99.940 ETH
71942.32 USD

>

345cA3...3e10

Amount

0 ETH
0.00 USD

Gas Limit

63207

UNITS

Gas Price

4

GWEI

Max Transaction Fee

0.000252 ETH
0.18 USD

Max Total

0.000252 ETH
0.18 USD


Data included: 36 bytes

RESET

SUBMIT

REJECT

Melissa




Breed: Boxer

Age: 2

Location: Camas, Pennsylvania

Adopt

Coleman



提交之后如果成功扣除了以太币，页面没有自动刷新的话，手动刷新下，发现"Adopt"按钮变成了"Success"。

Pete's Pet Shop

Private Network

wenzil

0x62730...

99.940 ETH
71942.20 USD

BUY SEND

SENT TOKENS


4 May 21 2018 01:31
0x345cA3e0...3e10

0 May 21 2018 01:28
0x2C2B9C9a...8FE4 (Failed)

0 May 21 2018 01:28
0x2C2B9C9a...8FE4 (Rejected)

0 May 21 2018 01:26
0x2C2B9C9a...8FE4 (Failed)

Melissa



Breed: Boxer
Age: 2
Location: Camas, Pennsylvania

Success

Coleman

备注:如果发现提交失败(如上图的"Failed"), 可以先检查IP和端口是否一致, 然后先关闭lite-server, 之后进入到"truffle develop"重新编译或者部署合约, 最后启动lite-server再进行测试。