

# 基于以太坊的投票系统

## 一、 开发环境搭建

### 1.1 安装 Go 环境。

```
Macbook:~ lisicong$ go version  
go version go1.11.2 darwin/amd64
```

### 1.2 安装 Node.js、Npm。

```
Macbook:~ lisicong$ node -v  
v10.14.1  
Macbook:~ lisicong$ npm -v  
6.4.1
```

### 1.3 安装 ganache-cli。

```
Macbook:~ lisicong$ ganache-cli  
Ganache CLI v6.2.5 (ganache-core: 2.3.3)
```

### 1.4 安装 truffle

```
Macbook:~ lisicong$ truffle v  
Truffle v4.1.15 (core: 4.1.15)  
Solidity v0.4.25 (solc-js)  
Macbook:~ lisicong$
```

## 二、 功能设计

作为一个用于自主选举的系统，本程序需要实现的功能如下：

- 1) 用户能够申请成为候选人。
- 2) 用户能够为所有候选人投票。
- 3) 能够统计所有候选人得到的票数。
- 4) 对候选人和投票人进行统计，包括人数、个人信息。

### 三、 合约设计

3.1 候选人与投票人的结构体，本结构体记录了个人信息。

```
//候选人的结构体
struct Candidate {
    bytes32 name; //候选人姓名
    bytes32 party; //候选人政党
    bool gender; //候选人性别, 1 male 0 female
    bool exist; //用于确认候选人存在
}

//投票者的结构体
struct Voter {
    bytes32 voterID; //投票者的 ID
    uint beVotedID; //被投候选人的 ID
}
```

3.2 为了储存结构体，制作投票人和候选人的哈希表。

```
//哈希表, key 为 uint, value 为结构体
mapping (uint => Candidate) candidates;
mapping (uint => Voter) voters;
```

3.3 添加获选人函数，包括候选人的姓名、政党（竞选宣言）、性别。

```
//添加候选人
function addCandidate(bytes32 name, bytes32 party, bool gender)
onlyOwner public {
    // 确认候选人 ID
    uint candidateID = numCandidates++;
    // 加入到结构体
    candidates[candidateID] = Candidate(name,party,gender,true);
    emit AddedCandidate(candidateID);
}
```

3.4 投票人为候选人投票

```
//投票
function vote(bytes32 voterID, uint candidateID) public {
```

```

        // 如果存在此候选人
        if (candidates[candidateID].exist == true) {
            // 加入到结构体
            voters[numVoters++] = Voter(voterID,candidateID);
        }
    }
}

```

### 3.5 统计候选人投票

```

// 遍历统计票数
function totalVotes(uint candidateID) view public returns (uint) {
    // 初始化
    uint res = 0;
    for (uint i = 0; i < numVoters; i++) {
        // 统计票数
        if (voters[i].beVotedID == candidateID) {
            res++;
        }
    }
    return res;
}

```

### 3.6 统计信息，包括统计投票人和候选人的所有信息。

```

//返回候选人数量
function getNumOfCandidates() public view returns(uint) {
    return numCandidates;
}

//返回投票人数量
function getNumOfVoters() public view returns(uint) {
    return numVoters;
}

//根据 name 返回候选人 ID
function getCandidate(bytes32 Name) public view returns (uint) {
    for (uint i = 0; i < numCandidates; i++) {

```

```

        if (candidates[i].name == Name) {
            return i;
        }
    }
    return 999;
}
//根据 ID 返回候选人信息
function getCandidate(uint ID) public view returns (uint, bytes32,
bytes32, bool) {
    return
(ID,candidates[ID].name,candidates[ID].party,candidates[ID].gender);
}
//返回投票人信息
function getVoter(uint ID) public view returns (uint,bytes32,uint) {
    return (ID, voters[ID].voterID,voters[ID].beVotedID);
}

```

#### 四、 系统实现与部署

首先进行 app.js 的编写。

##### 4.1 初始化，加入两个候选人供选择

```

//初始化
start: function() {
    // 提供合约支持
    VotingContract.setProvider(window.web3.currentProvider)
    VotingContract.defaults({from:
window.web3.eth.accounts[0],gas:6721975})
    VotingContract.deployed().then(function(instance){

        // 调用 getNumOfCandidates()
        instance.getNumOfCandidates().then(function(numOfCandidates){

            // 加入新建候选人

```

```

        if (numOfCandidates == 0){

instance.addCandidate("Frank","Democratic",true).then(function(result) {
    $("#candidate-box").append('<div class='form-check'><input
class='form-check-input' type='checkbox' value=''
id=${result.logs[0].args.candidateID}><label class='form-check-label'
for=0>Frank</label></div>`)
    })

instance.addCandidate("Sicilia","Republican",false).then(function(result) {
    $("#candidate-box").append('<div class='form-check'><input
class='form-check-input' type='checkbox' value=''
id=${result.logs[0].args.candidateID}><label class='form-check-label'
for=0>Sicilia</label></div>`)
    })
    numOfCandidates = 2
}
//如果有候选人，显示
else {
    for (var i = 0; i < numOfCandidates; i++ ){
        instance.getCandidate(i).then(function(data) {
            $("#candidate-box").append('<div class="form-check"><input
class="form-check-input" type="checkbox" value="" id=${data[0]}><label
class="form-check-label"
for=${data[0]}>${window.web3.toAscii(data[1])}</label></div>`)
            })
        }
    }
    window.numOfCandidates = numOfCandidates
})
}).catch(function(err){
    console.error("ERROR! " + err.message)
}

```

```
    })  
  },
```

#### 4.2 申请成为候选人

//参选

```
beCandidate: function() {  
  var name = $("#name-input").val()  
  var party = $("#party-input").val()  
  var gender = $("#gender-input").val()  
  if(gender == "male") gender = true  
  else gender = false  
  if (name == "" || party == "" || gender == "") {  
    $("#msg2").html("<p>Please enter information.</p>")  
    return  
  }  
  VotingContract.deployed().then(function(instance) {  
    instance.addCandidate(name, party, gender)  
    //刷新候选人数量  
    window.numOfCandidates = numOfCandidates  
  })  
  for (var i = 0; i < numOfCandidates; i++) {  
    instance.getCandidate(i).then(function(data) {  
      $("#candidate-box").append('<div class="form-check"><input  
class="form-check-input" type="checkbox" value="" id=${data[0]}><label  
class="form-check-label"  
for=${data[0]}>${window.web3.toAscii(data[1])}</label></div>')  
    })  
  }  
},
```

#### 4.3 投票人进行投票

// 投票

```

vote: function() {
  var uid = $("#id-input").val() //获取 voterID
  if (uid == ""){
    $("#msg").html("<p>Please enter ID.</p>")
    return
  }
  if ($("#candidate-box :checkbox:checked").length > 0){
    //获取候选人 ID
    var candidateID = ($("#candidate-box :checkbox:checked")[0].id
  }
  else {
    //如果没有选择
    $("#msg").html("<p>Please choose a candidate.</p>")
    return
  }
  // 显示投票成功
  VotingContract.deployed().then(function(instance){
    instance.vote(uid,parseInt(candidateID)).then(function(result){
      $("#msg").html("<p>Voted</p>")
    })
  }).catch(function(err){
    console.error("ERROR! " + err.message)
  })
},

```

代码完成之后开始部署合约。

1. 首先编译合约 (truffle compile)，如下图所示：

```

Macbook:~ lisicong$ cd Desktop/voting/
Macbook:voting lisicong$ truffle compile
Compiling ./contracts/Voting.sol...

Compilation warnings encountered:

/Users/admin/Desktop/voting/contracts/Voting.sol:9:5: Warning: Defining constructors as functions with the same name as the contract is deprecated. Use "constructor(...) { ... }" instead.
    function Voting()public {
      ^ (Relevant source part starts here and spans across multiple lines).

Writing artifacts to ./build/contracts

```

得到编译成功文件。

然后新建一个命令窗口，开启以太坊私链（ganache-cli），如下图所示：

```

(8) 0x8318051b9d84d5612ebe5012654c8fa05a7cdc9ec609306b9b7af0833405e2aa
(9) 0x717b194b55a04138915cb9e828938e7526055ef946181ec9dc2de3ac23d43ebe

HD Wallet
=====
Mnemonic:      always girl hospital virus dash fossil sponsor junior base nice p
act wheel
Base HD Path:  m/44'/60'/0'/0/{account_index}

Gas Price
=====
20000000000

Gas Limit
=====
6721975

Listening on 127.0.0.1:8545
eth_blockNumber
eth_blockNumber
eth_blockNumber
eth_blockNumber

```

然后在原窗口进行合约部署（truffle migrate），如下图所示：



```

Macbook:voting lisicong$ truffle migrate
Compiling ./contracts/Voting.sol...

Compilation warnings encountered:

/Users/admin/Desktop/voting/contracts/Voting.sol:9:5: Warning: Defining constructors as functions with the same name as the contract is deprecated. Use "constructor(...) { ... }" instead.
    function Voting()public {
    ^ (Relevant source part starts here and spans across multiple lines).

Writing artifacts to ./build/contracts

Using network 'development'.

Running migration: 1_initial_migration.js
  Deploying Migrations...
  ... 0x1d503e39cc4fc4a8b5b15e3f61b98a0fa58e386373ef4cbb203701b3b3bf6278
  Migrations: 0x309df2e907b9b29db24045bf64d2dd97d5749773
Saving successful migration to network...
  ... 0x0eda7996c67927f318d53151cdf4a8eb9c2c0a4b1e05ebe5eb0d8704bae31ea
Saving artifacts...
Running migration: 2_deploy_contracts.js
  Deploying Voting...
  ... 0xbf7f133bd0f4dd506b2ffe703e98715ec196f756768754aeb2834dd82dd13baf
  Voting: 0x89349330cfb21233a1274449d06f23f944ce372d
Saving successful migration to network...
  ... 0x3eb73a1109f679066471ce18b74efe3cc165bde9b72b82d3c39e27a87f31fa06
Saving artifacts...

```

得到合约的地址。

然后开启服务器，如下图所示：

```

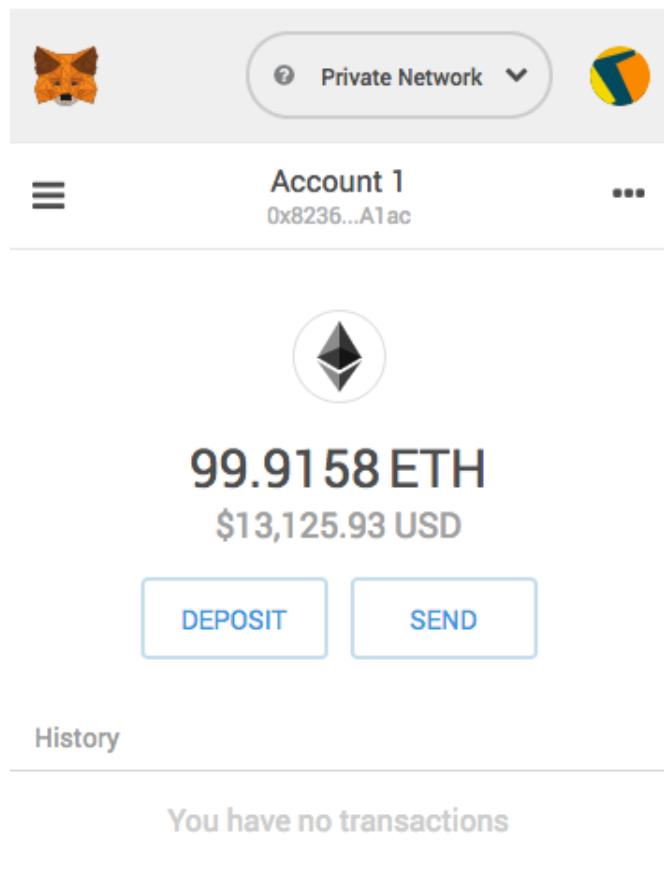
Macbook:voting lisicong$ npm run dev

> ethereum-voting-dapp@1.0.0 dev /Users/admin/Desktop/voting
> webpack-dev-server

Project is running at http://localhost:8080/
webpack output is saved to ...

```

打开浏览器界面进行 Metamask 的配置：



之后就可以进入 127.0.0.1:8000 查看部署的合约了。

## 五、 主要界面截图与说明

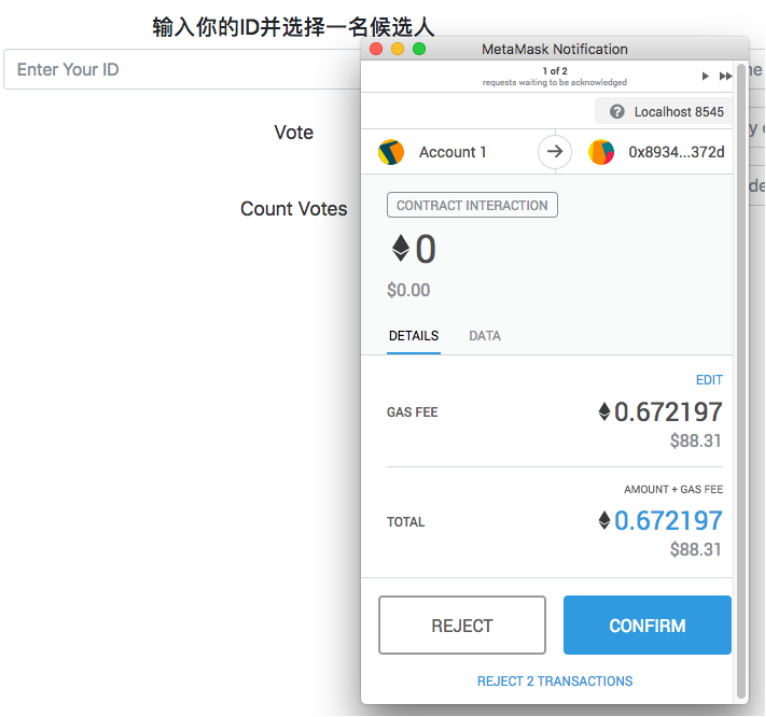
进入网页，首先看到一个没有候选人的投票系统。

### 基于以太坊的投票系统

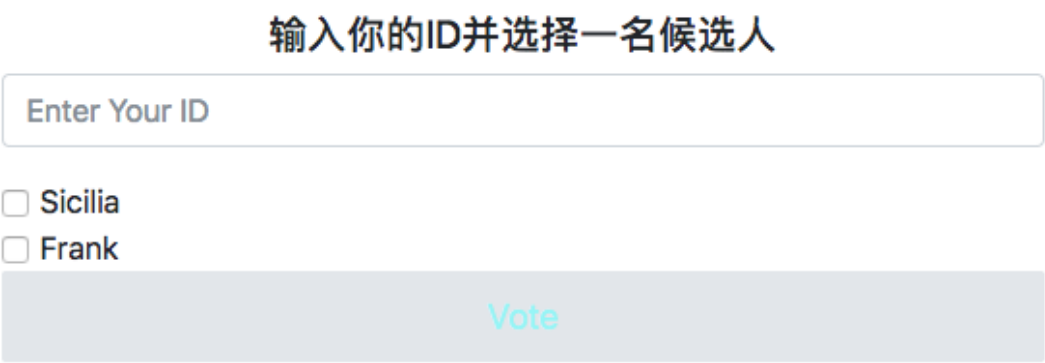
输入你的ID并选择一名候选人	输入你的信息并参加竞选
<input type="text" value="Enter Your ID"/>	<input type="text" value="Enter Your Name"/>
<input type="button" value="Vote"/>	<input type="text" value="Enter Your Party or Declaration"/>
<input type="button" value="Count Votes"/>	<input type="text" value="Enter Your Gender"/>
	<input type="button" value="参选"/>

支付一定以太币后，可以看到初始化的两个候选人，下图为支付信息。

# 基于以太坊的投票系统



如下图所示，出现了两个预加的候选人。



之后，选择申请成为一个候选人，名为 test：

## 输入你的信息并参加竞选

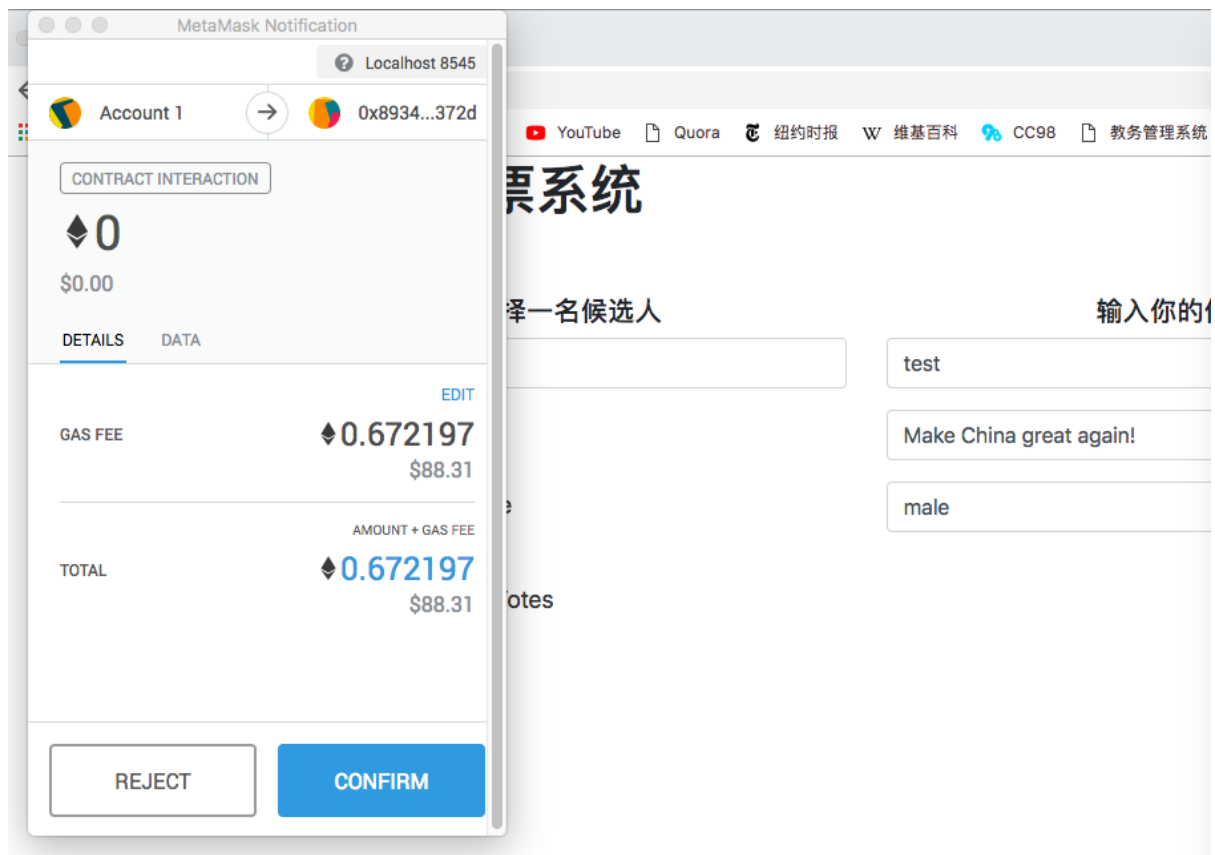
test

Make China great again!

male

参选

然后支付一定的以太币：



如下图所示，刷新界面，可以看到 test 已经加入候选人名单了。

## 输入你的ID并选择一名候选人

- ☐ Sicilia
- ☐ Frank
- ☐ test

Vote

## Count Votes

之后，给其中一位候选人投票：

## 输入你的ID并选择一名候选人

- ☒ Sicilia
- ☐ Frank
- ☐ test

Vote

如下图所示，Sicilia 得到了一票。

Count Votes

Sicilia: 1

Frank: 0

test: 0