

# Part 2 – Code and Deploy a smart contract on Ethereum testnet “Ropsten”

**IMPORTANT** : At the end of your exercise push the code of the smart contract on your created github project “Blockchain distributed system & Smart contract” where you create a new repository name “Tutorial 2 : Part 2”.

When you finished to complete the Quest please provide your github account with all files (configurations, smart contract code ...) created for this tutorial including this tutorial;

- Public key of your metamask account
- All the transactions ID generated
- Send me 1 ETH on my Ethereum Wallet on ropsten testnet with the metamask account used for this tutorial :

0xddF8CDa9d8A425b5a8FFb64BfC79857bA9b57083

You have to create also a report with all print screens for each following steps.

## Pré-requis :

Code du Smart contract à déployer : <https://github.com/cozcan/KamalRajniElection>

Metamask un wallet intégrer dans votre navigateur Chrome [.metamask.io](https://metamask.io) et détenir des Ethers sur Ropsten

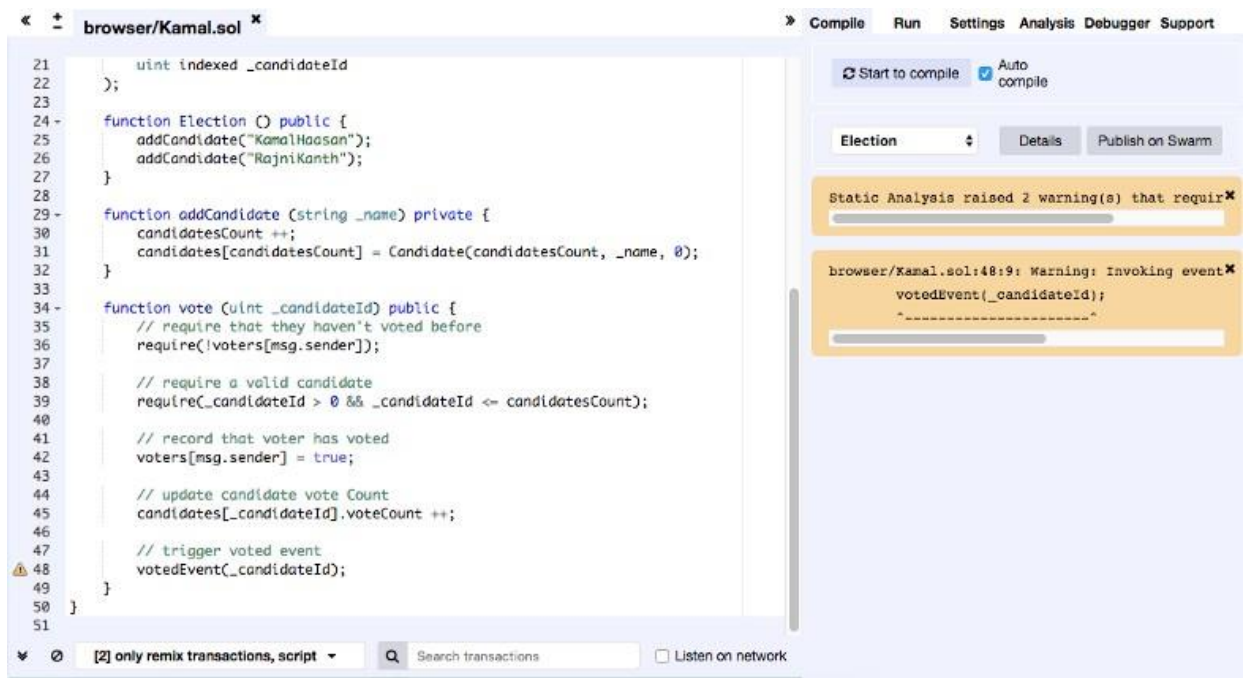
Remix : [Remix - Solidity IDE](https://remix.ethereum.org) disponible via votre navigateur [remix.ethereum.org](https://remix.ethereum.org)

Ropsten Test Network : Environment de test

---

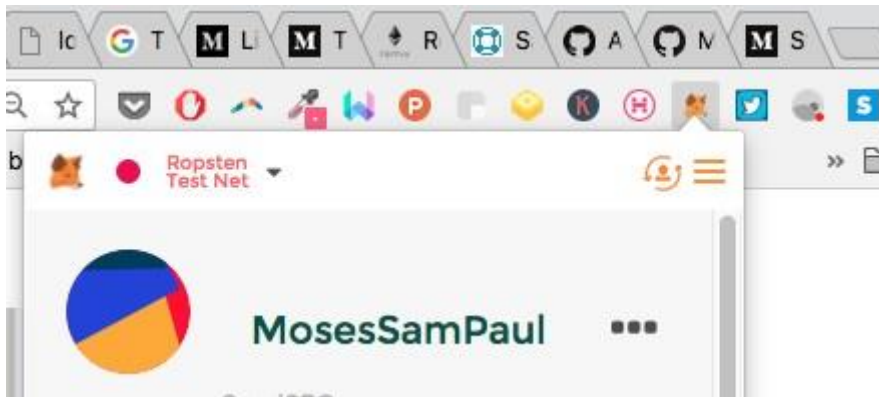
## Stage 1: Setup the Environment

Copy the election.sol (smart contract) and paste it into the remix IDE.

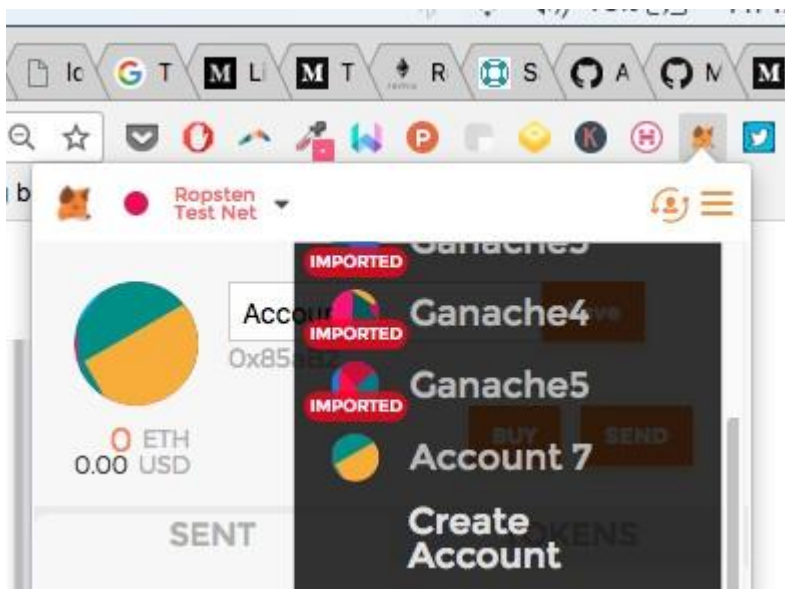


To deploy the contract, we need an account and with some ether on the Ropsten test net.

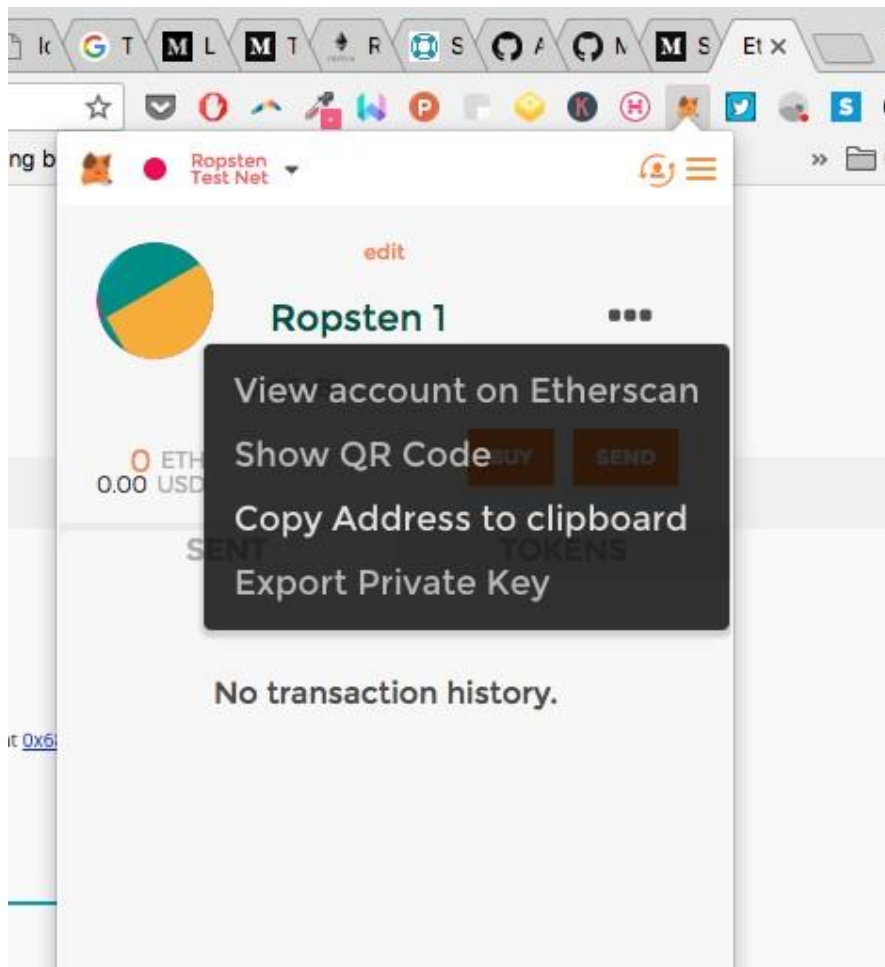
- Meta mask!
- Select Ropsten Test net



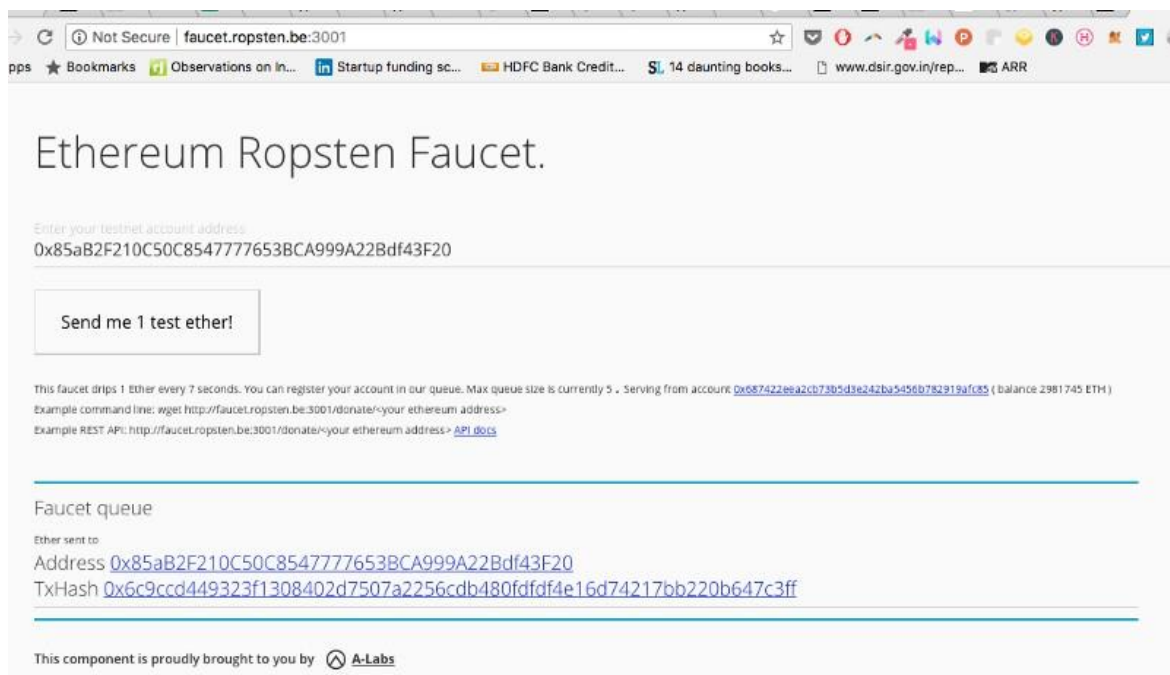
- Create an Account



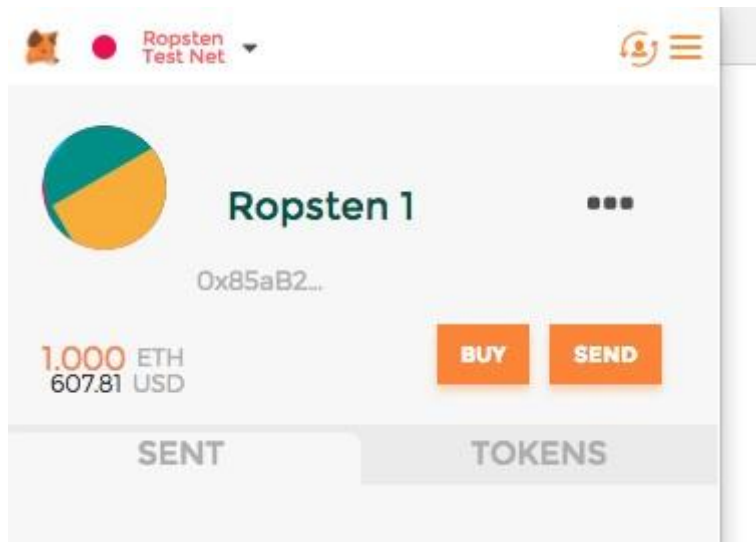
- Copy Account Address



- Request Free ether from [here](#) or [here](#). Paste the copied account address in text box and click on send me 1 test ether.



- Ogle at your free ether :P



So now, we have the contract pasted in remix IDE, we have connected meta mask to the ropsten testnet with an account that has some ether.

## Stage 2: Deploy the contract

- click on Run. Select Injected Web 3 Ropsten under environment and the account in Metamask is shown here under Account with balance ether as well.

```
1 pragma solidity ^0.4.2;
2
3 contract Election {
4     // Model a Candidate
5     struct Candidate {
6         uint id;
7         string name;
8         uint voteCount;
9     }
10
11     // Store accounts that have voted
12     mapping(address => bool) public voters;
13     // Store Candidates
14     // Fetch Candidate
15     mapping(uint => Candidate) public candidates;
16     // Store Candidates Count
17     uint public candidatesCount;
18
19     // voted event
20     event votedEvent (
```

Environment: Injected Web3 Ropsten (3) ⓘ

Account: 0x063...7ee03 (4.9707214 ether) ⓘ

Gas limit: 3000000

Value: 0 wei

Election ⓘ

Deploy

or

At Address Load contract from Address

- Click on Deploy -> confirm transaction

The screenshot displays the Remix IDE interface during the deployment of a Solidity contract named 'Election'.

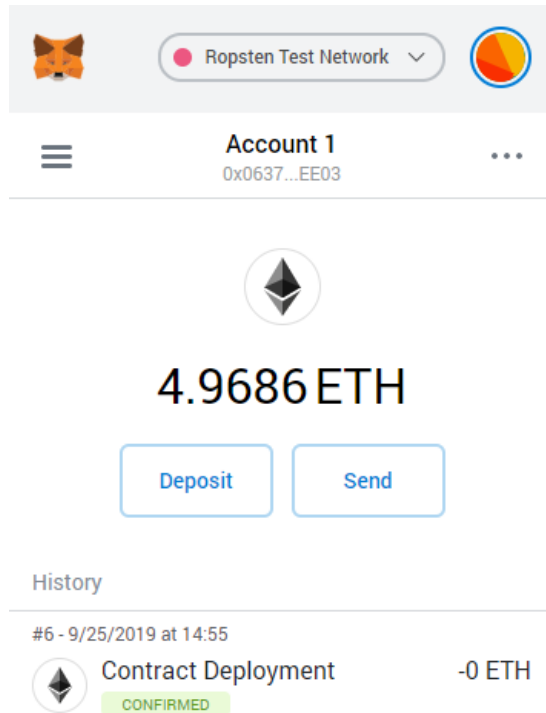
**Left Panel (Editor):** Shows the Solidity code for the 'Election' contract. The code includes a pragma statement for Solidity version 0.4.2, a struct for 'Candidate' with fields 'id', 'name', and 'voteCount', and functions for 'addCandidate' and 'vote'.

**Center Panel (Contract Deployment):** Displays the 'CONTRACT DEPLOYMENT' window. It shows the account 'Account 1' with a balance of 0 ETH. The gas fee is 0.001066 ETH, and the total cost is 0.001066 ETH. The window includes a 'Reject' button and a 'Confirm' button.


**Right Panel (Run):** Shows the 'Run' tab with the 'Election' contract selected. It displays the environment 'Injected Web3' on the 'Ropsten (3)' network. The account address is '0x063...7ee03' with a balance of 4.96965549 ether. The gas limit is set to 3000000 and the value is 0 wei. The 'Deploy' button is visible.

**Bottom Panel (Console):** Shows the output of the deployment process, including the web3 version (1.0.0), ethers.js, and the compilers used.

- Click on Metamask extension!



- Click on the contract deployment — it should open up the etherscan page to look at our transaction details. Check the status is successful



Ropsten Testnet Network

All Filters

Search by Address / Txn Hash / Block / Token / Ens

🔍

Home


Blockchain

Tokens

Misc

Ropsten

### Transaction Details

Sponsored:  Allinfra brings access choice & liquidity to unlisted infrastructure assets. [Register today](#)

Overview

State Changes

[ This is a Ropsten Testnet transaction only ]

Transaction Hash:

0xc5e1f9c514eb5003fdc8c02546e011b48666782496bbcab9e44acd10b8f6f639

Status:

Success

Block:

6456751

11 Block Confirmations

Timestamp:

3 mins ago (Sep-25-2019 12:56:26 PM +UTC)

From:

0x0637f9a391f440599f1afb80540af2bdc217ee03

To:

[Contract 0x4b46ba4e598e5f700fa8bab8b234571eca71937 Created]

Value:

0 Ether (\$0.00)

Transaction Fee:

0.00106591 Ether (\$0.000000)

Click to see More

congrats, you have successfully created a smart contract

---

### Stage 3: Interact with the contract



Environment
Injected Web3
Ropsten (3)
i

Account
0x063...7ee03 (4.96858958 ether)
i

Gas limit
3000000

Value
0
wei

Election
i

Deploy

or

At Address
Load contract from Address

Transactions recorded: 3

Deployed Contracts

Election at 0xe62...e234f (blockchain)

addCandidate
string\_name

vote
uint256\_candidateId

candidates
uint256

candidatesCount

voters
address

- Pull / Access data from contract— free — shaded “blue”

If you add 1 (candidate ID) near the candidates field ..you should get the details of candidate ID 1. And 2 would give you the second candidate details.

Deployed Contracts

Election at 0xe62...e234f (blockchain)

addCandidate

string \_name

▼

vote

uint256 \_candidateId

▼

candidates

"1"

▼

0: uint256: id 1  
1: string: name KamalHaasan  
2: uint256: voteCount 0

candidatesCount

voters

address

▼

Deployed Contracts

Election at 0xe62...e234f (blockchain)

addCandidate

string \_name

▼

vote

uint256 \_candidateId

▼

candidates

"2"

▼

0: uint256: id 2  
1: string: name RajniKanth  
2: uint256: voteCount 0

candidatesCount

voters

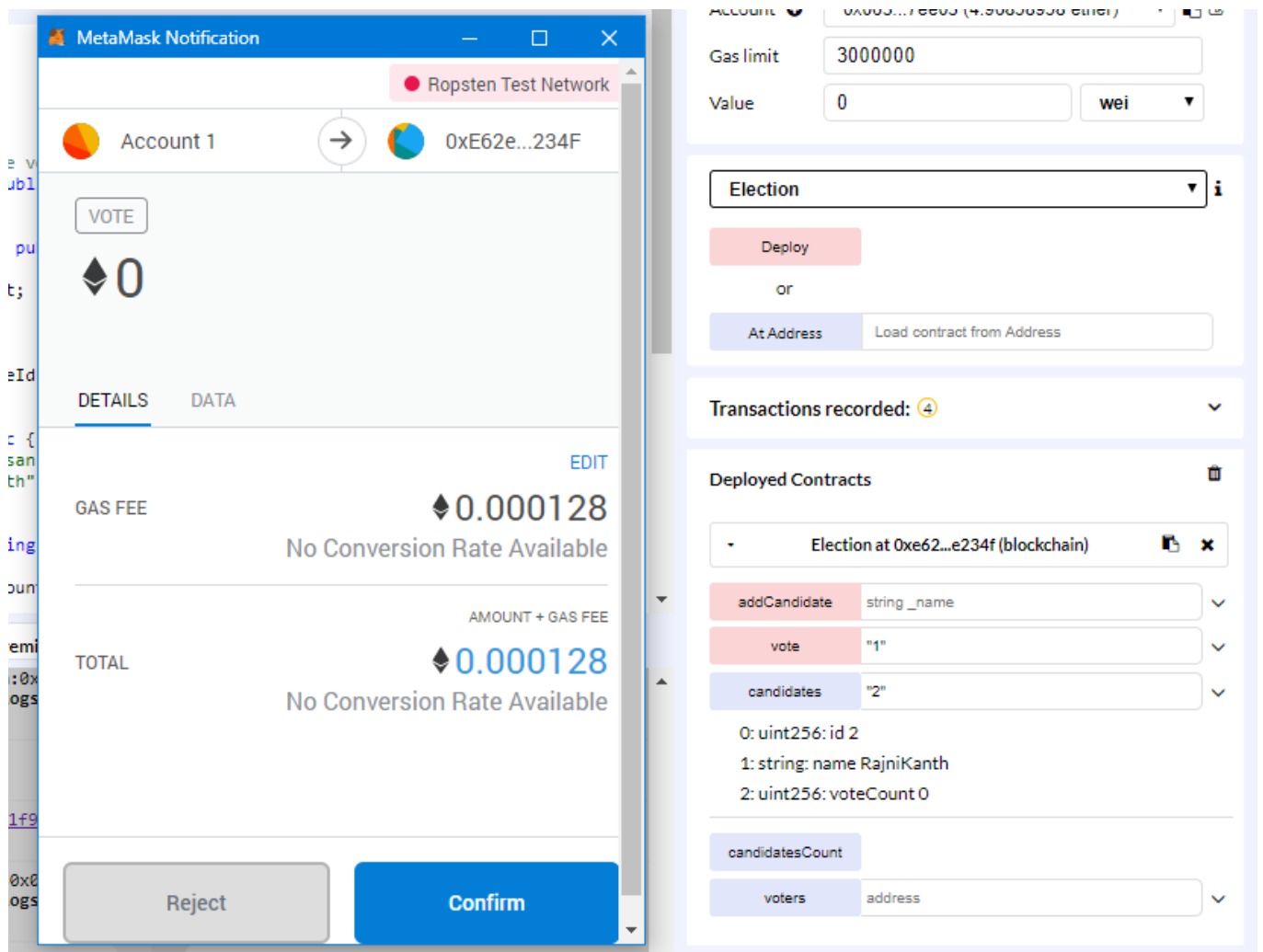
address

▼

remember pulling data is free on ethereum blockchain

- **Vote / add data to the chain!**

Now that we know who is whom. Let's try and Vote.

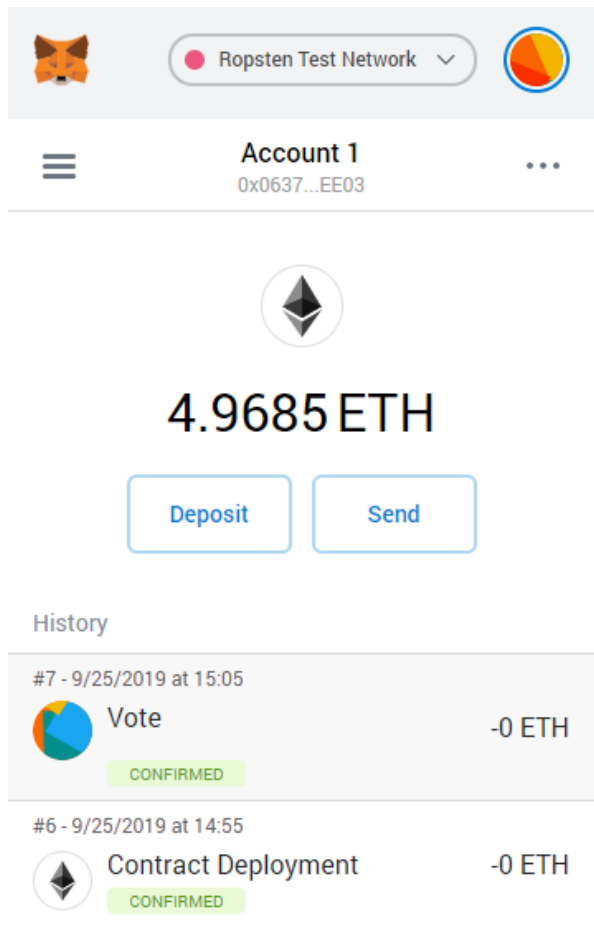


notice the 1 next to vote

transact to Election.vote pending ...

<https://ropsten.etherscan.io/tx/0xdfd8ada1e46357199fb9187a42bbc622799f5409d1336ecea3b18d9c70562c45>

IDE shows my transaction is pending



after few seconds, the Metamask extension shows me the transaction details. If I click on it, it should take me to ether scan.

Transaction successful!

Etherscan

Ropsten Testnet Network

All Filters Search by Address / Txn Hash / Block / Token / Ens

Home Blockchain Tokens Misc Ropsten

### Transaction Details

Sponsored: Diligence. The top auditors and security services providers you've been looking for. [Learn more](#)

Overview

Event Logs (1)

State Changes

[ This is a Ropsten Testnet transaction only ]

Transaction Hash:	0xdfd8ada1e46357199fb9187a42bbc622799f5409d1336ecea3b18d9c70562c45
Status:	Success
Block:	6456808 8 Block Confirmations
Timestamp:	2 mins ago (Sep-25-2019 01:06:36 PM +UTC)
From:	0x0637f9a391f440599f1afb80540af2bdc217ee03
To:	Contract 0xe62ea5d7ae07e4130d1b2ea7f14070d4e06e234f
Value:	0 Ether (\$0.00)
Transaction Fee:	0.000128148 Ether (\$0.000000)

[Click to see More](#)

If I copy my address from metamask and paste it into etherscan I can see all the transactions within etherscan

Address

0x85aB2F210C50C854777653BCA999A22Bdf43F20

Home / Normal Accounts / Address

Overview

ETH Balance:

1.991761258 Ether

No Of Transactions:

6 txns

Transactions

Latest 6 txns

TxHash	Block	Age	From		To	Value	[TxFee]
0x06d376e65fc00b...	2845023	4 mins ago	0x85ab2f210c50c85...	OUT	0xd3052b1eff78929...	0 Ether	0.0002958298
0x427ed64191ed59...	2844991	11 mins ago	0x85ab2f210c50c85...	OUT	Contract Creation	0 Ether	0.000308391
0xd1cf326b054oodf...	2844900	36 mins ago	0x85ab2f210c50c85...	OUT	Contract Creation	0 Ether	0.001299818
0x8fc80f913e86472f...	2844900	36 mins ago	0x85ab2f210c50c85...	OUT	Contract Creation	0 Ether	0.001299818
0x66b6ca5101a8f6...	2844886	39 mins ago	0x81b7e08f65bdf56...	IN	0x85ab2f210c50c85...	1 Ether	0.00021
0x6c9ccd449323f13...	2844845	49 mins ago	0x687422eea2cb73...	IN	0x85ab2f210c50c85...	1 Ether	0.00021

The Green INs are the ethers that have come to me from the faucets.  
The Orange OUTs are my spends. I have created the contract thrice and wrote on it once.

If I click on the first line item the vote transaction, this is what we see in etherscan.