Building Smart Contracts with Remix

NINA BREZNIK
@ninabreznik

YANN LEVREAU
@ninabreznik

IURI MATIAS
@iurimatias

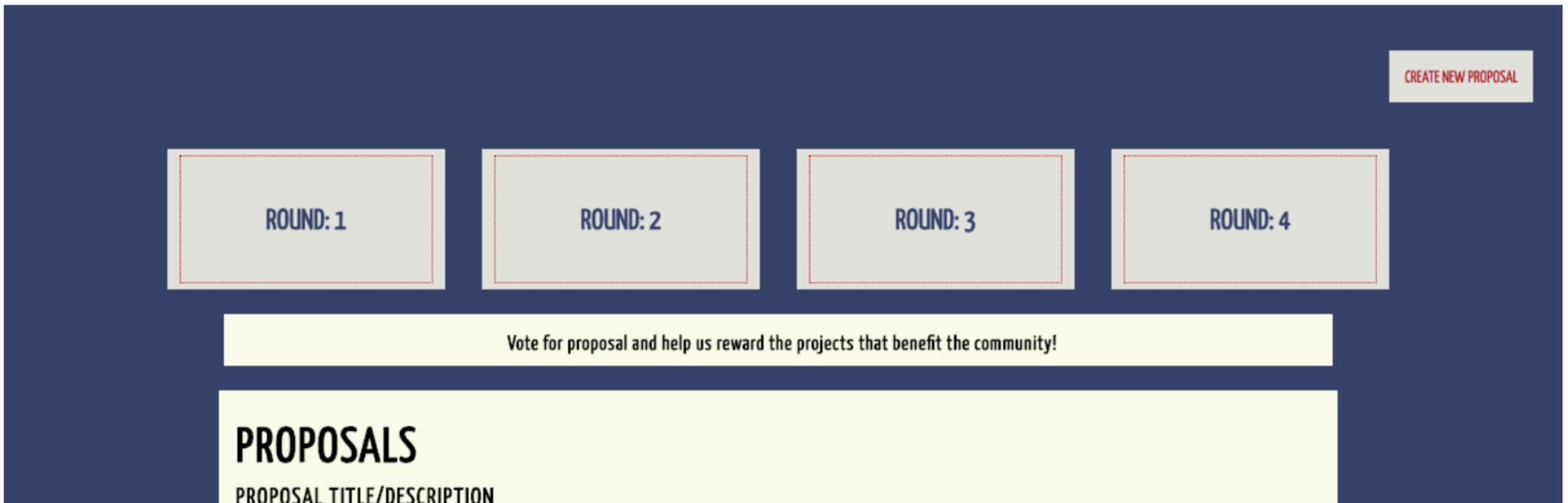
ROB STUPAY
@ryestew

ALEX PRAETORIUS
@SERAPATH

Ballot Dapp Workshop

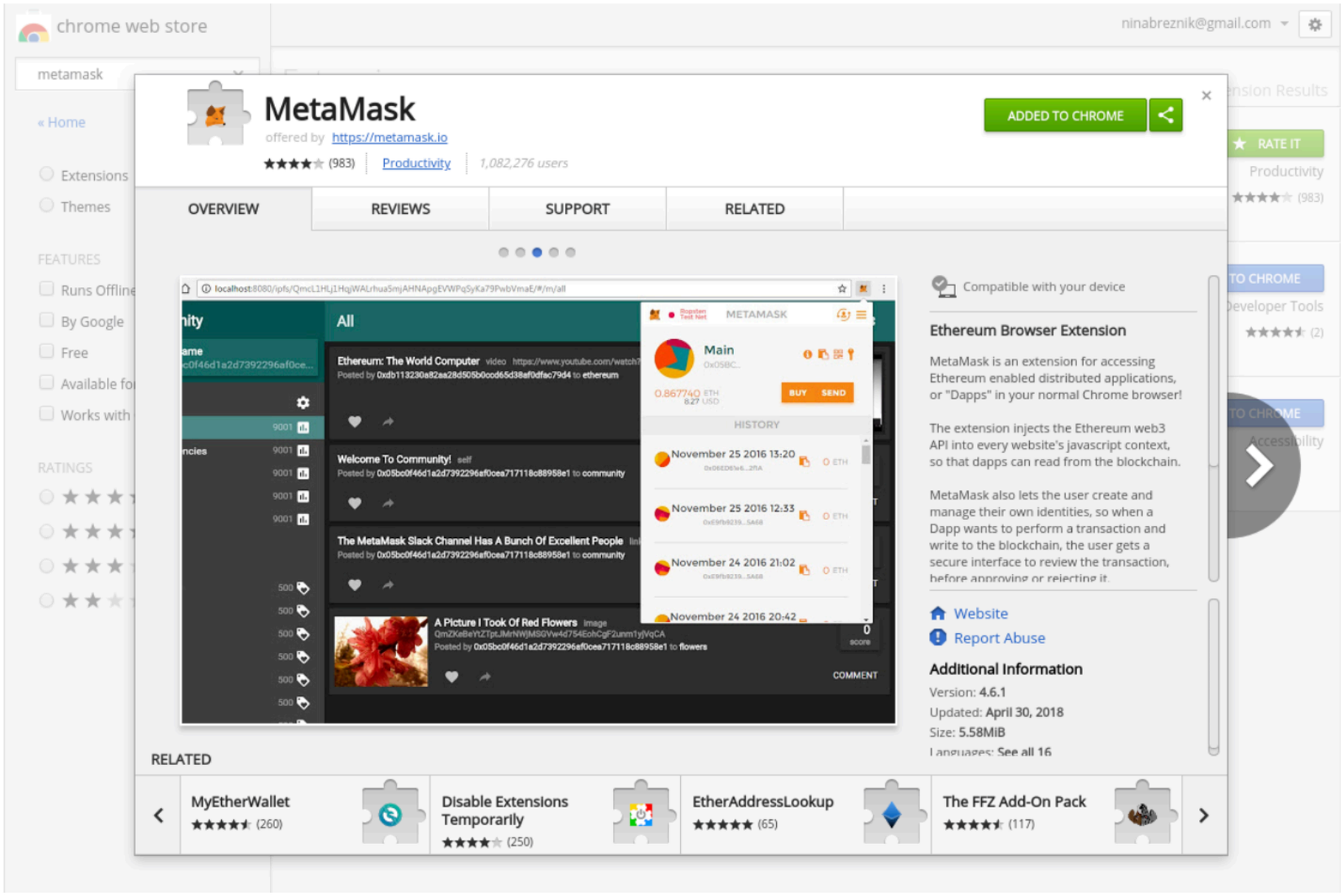
bit.ly/remix-workshop-repository

PDF: https://updig.is/pdf/remix-chez-coinhouse.pdf



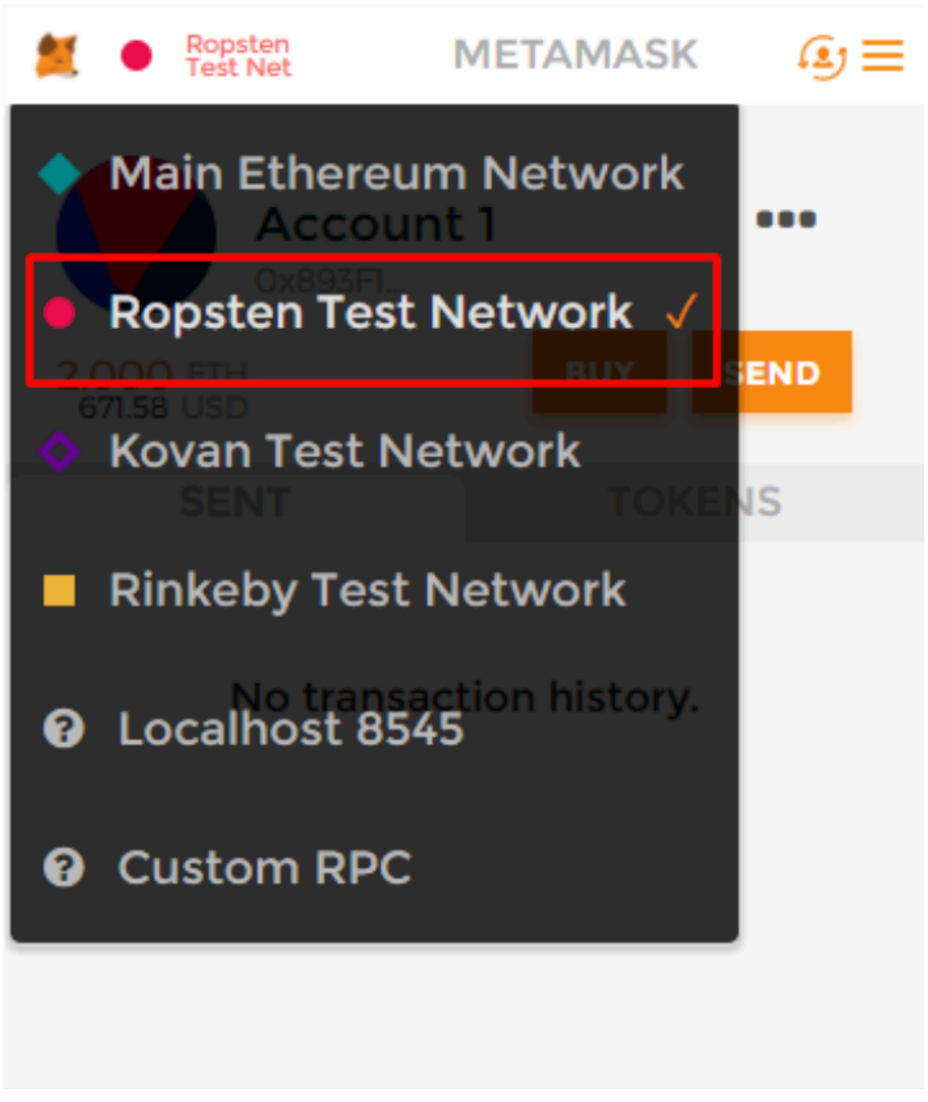
Install Metamask

chrome.google.com/webstore



Login to Metamask

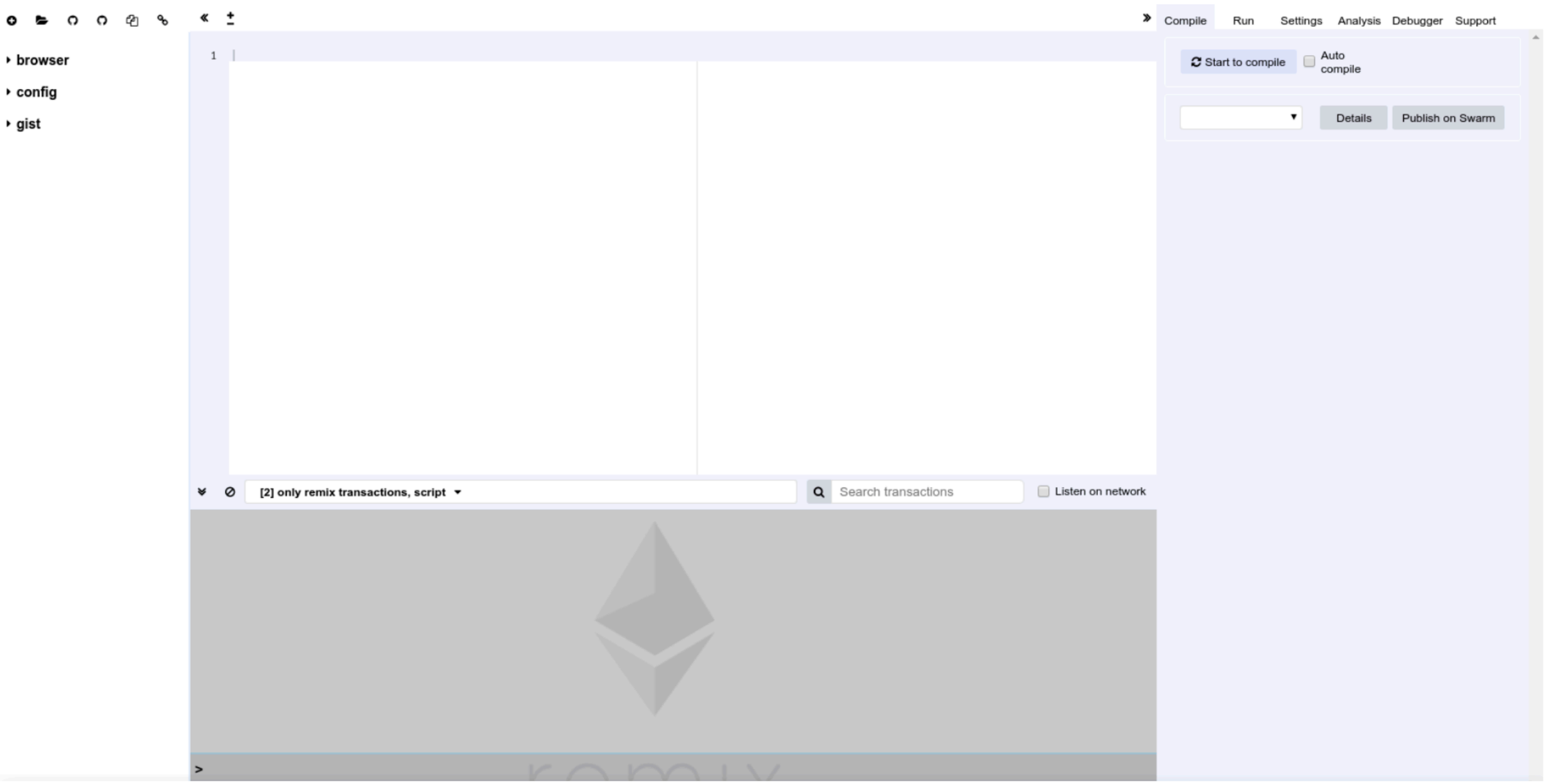
Robsten Test Network



Let’s get started

https://bit.ly/remix-workshop-repository

https://remix-alpha.ethereum.org

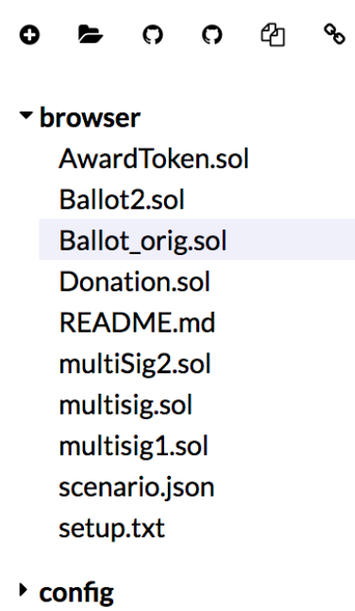


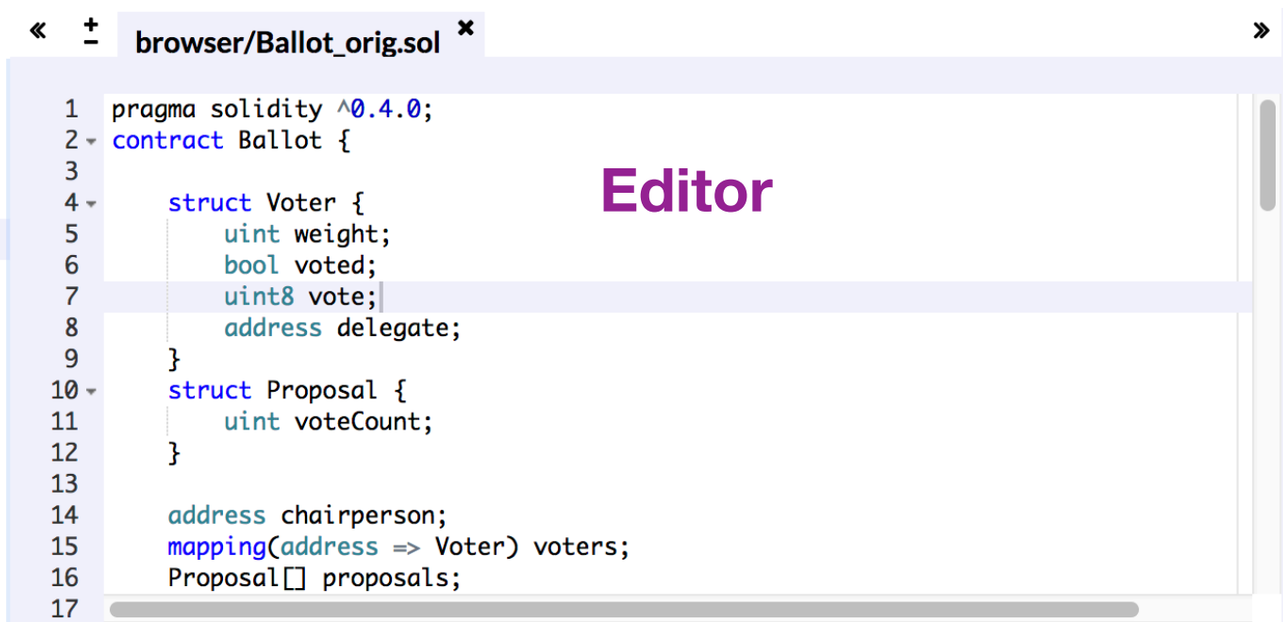
Remix Tour

File Explorer

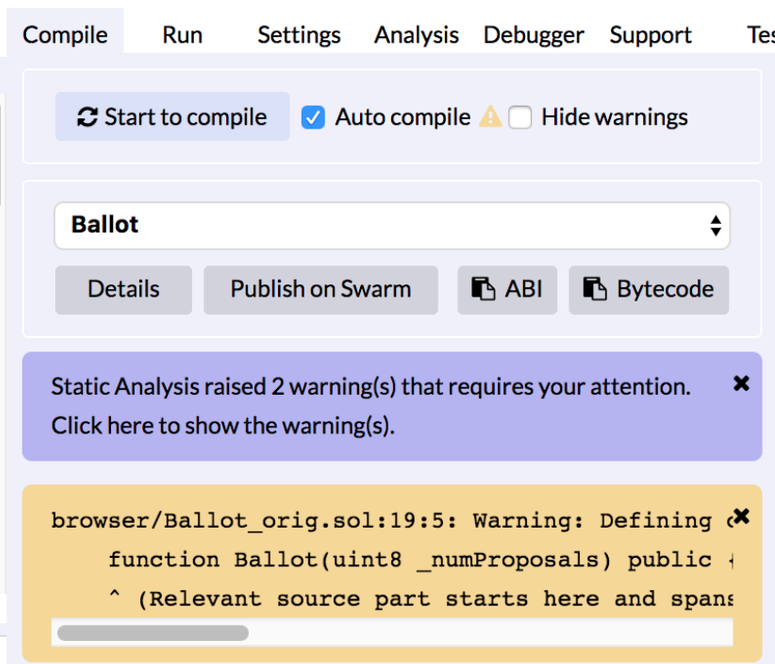
https://remix-alpha.ethereum.org

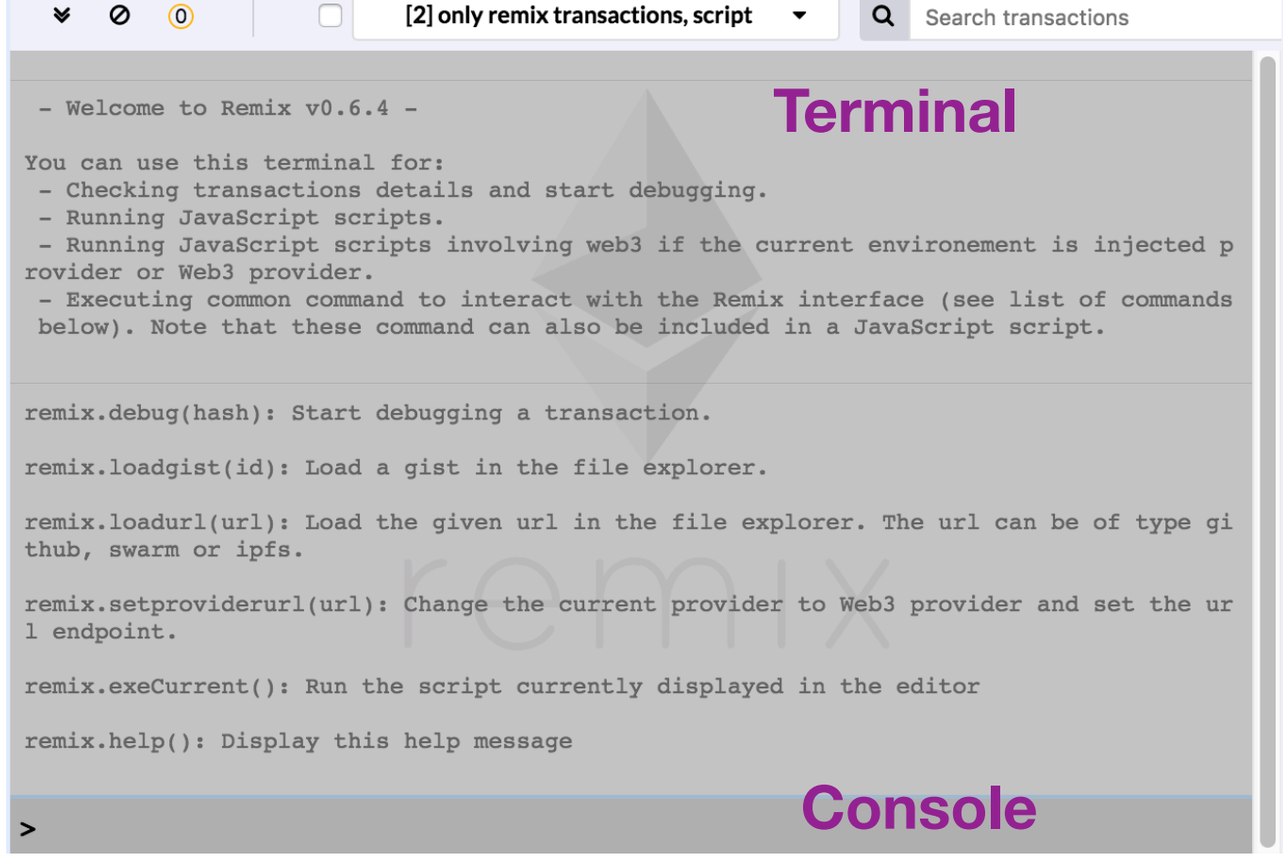
Compile Tab (active)





Editor





Terminal

Console

Run Tab

Compile | Run | Settings | Analysis | Debugger | Support | Test

Environment: Injected Web3 — Ropsten (3)

Account: 0x9ae...06ff6 (1.992485469305616838

Gas limit: 3000000

Value: 0 wei

AwardToken

Deploy

Load contract from Address | At Address

Transactions recorded: 4

Deployed Contracts

AwardToken at 0x574...40360 (blockchain)

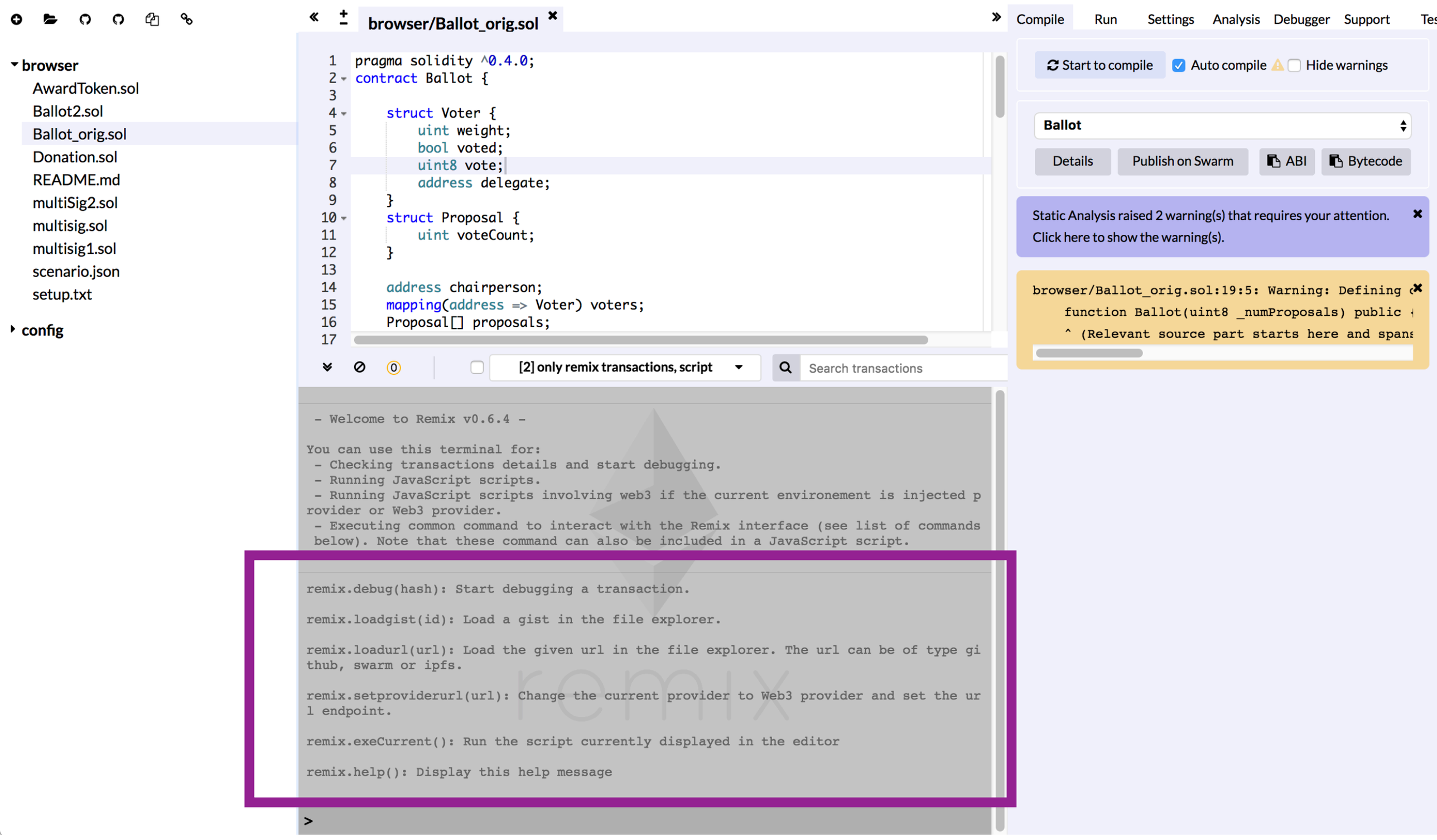
approve	address _spender, uint256 _value
closeRound	
closeRoundEarly	
decreaseApproval	address _spender, uint256 _subtractedValue
finishMinting	
increaseApproval	address _spender, uint256 _addedValue
mint	address _to, uint256 _amount

Universal DAPP

UI to the Contract

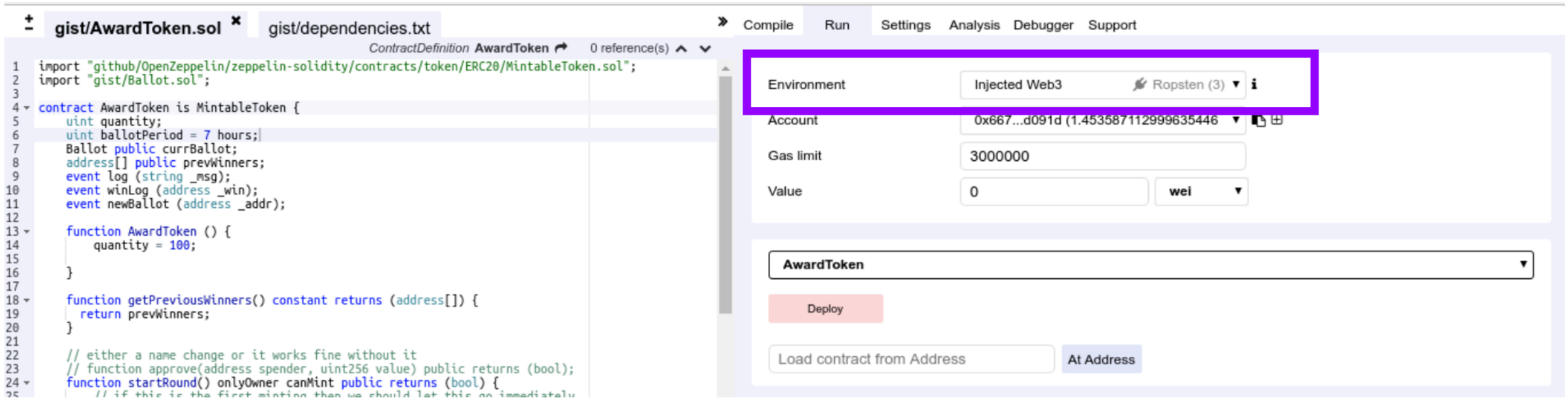
Remix Commands

https://remix-alpha.ethereum.org



Set environment

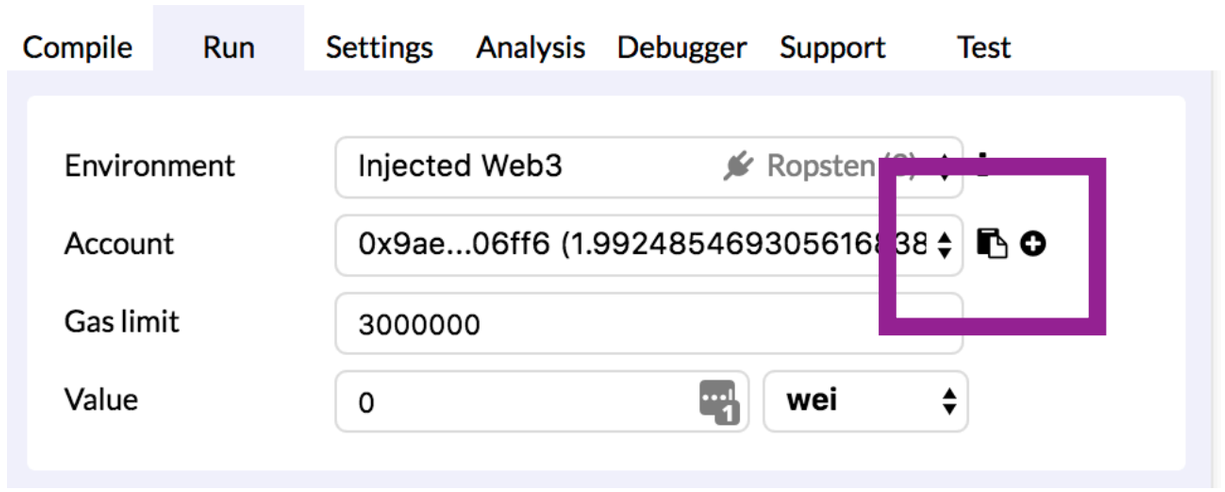
Run tab: Environment = Injected web3
(Ropsten)



Get some TEST ether

http://faucet.ropsten.be:3001/

But FIRST:
Copy your address - here or in Metamask



Load files to Remix

remix.loadgist('add9633c2b0101f6fda0aadcfe350f60')

```
24    // either a name change or it works fine without it
25    // function approve(address spender, uint256 value) public returns (bool);
```

[2] only remix transactions, script

Search transactions

remix

```
> remix.loadgist('2bf3c8833eac0814cf3aa1672217b6fb')
```

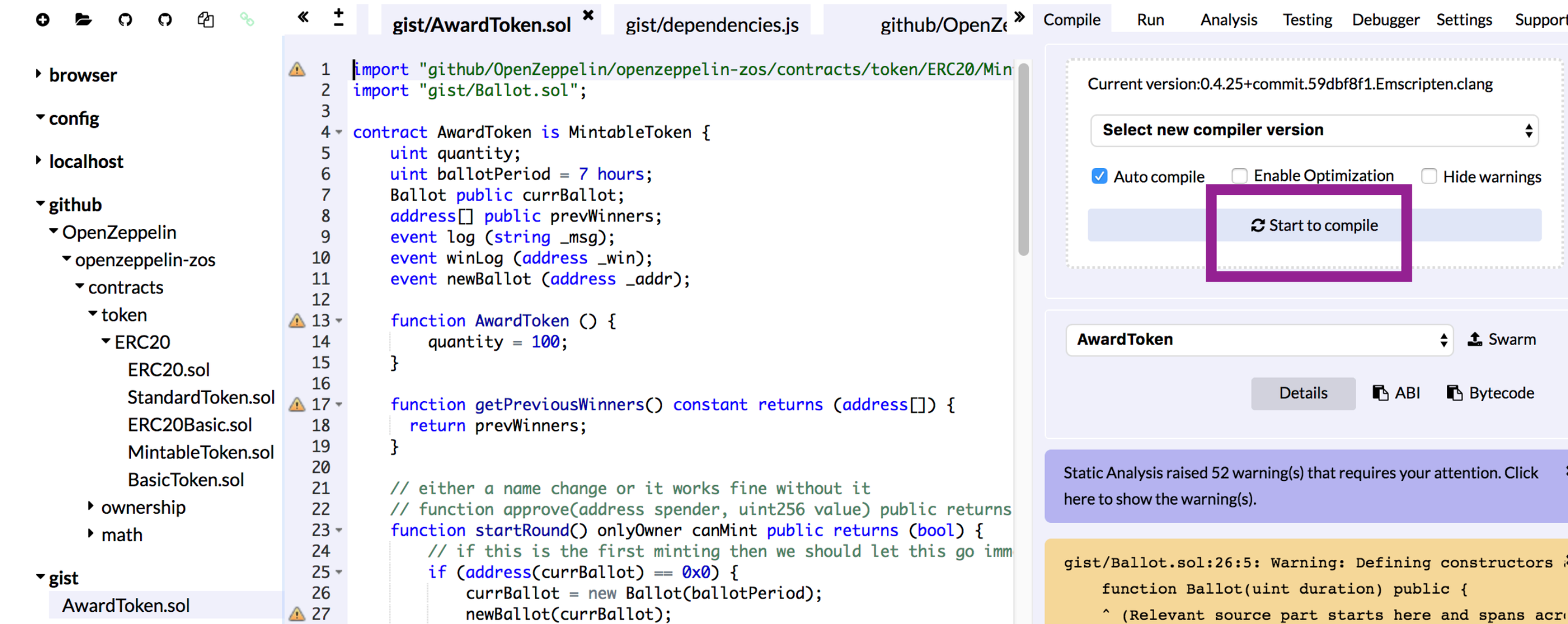
here in the console

Compile AwardToken

Make sure that AwarkToken.sol is the active file in the terminal.

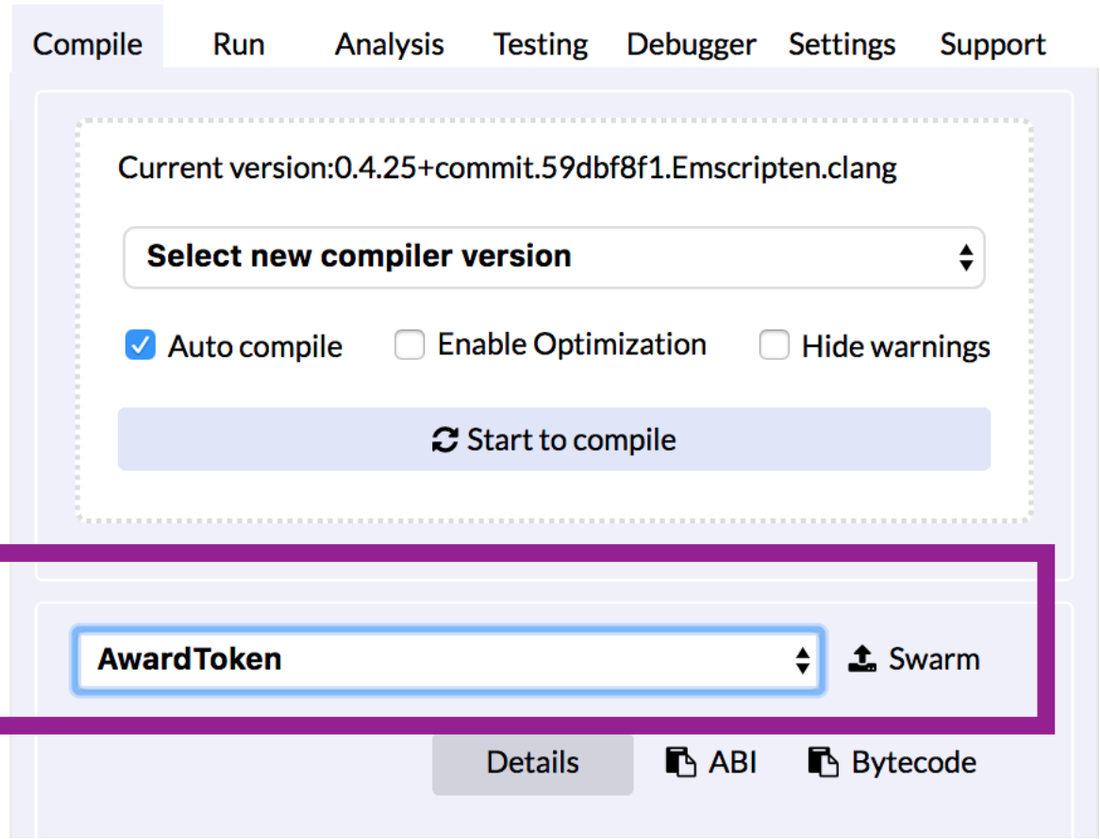
Click the “Start to compile” button

Update image w/o github

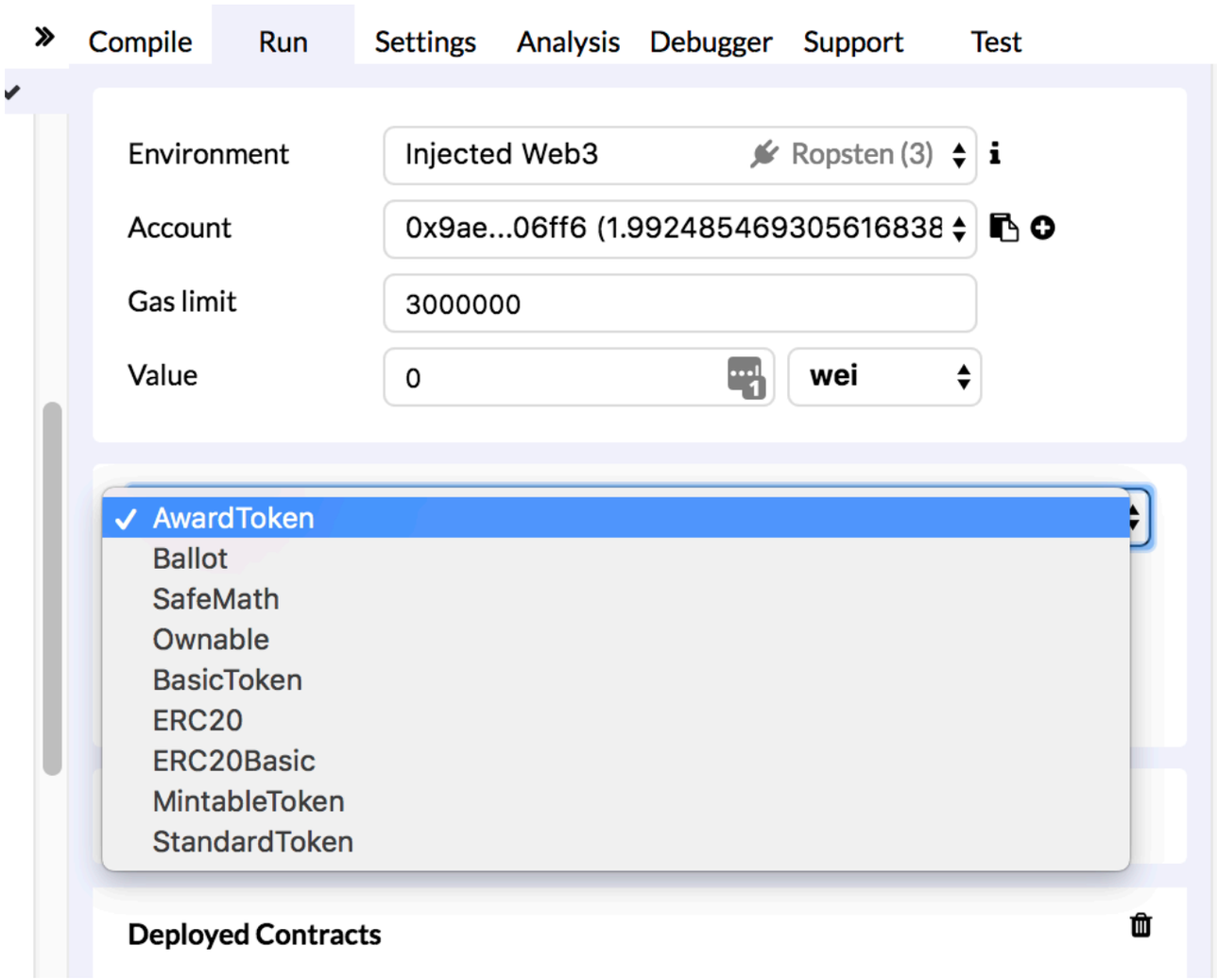


See compiled contracts

AwardToken + all it's imported contracts

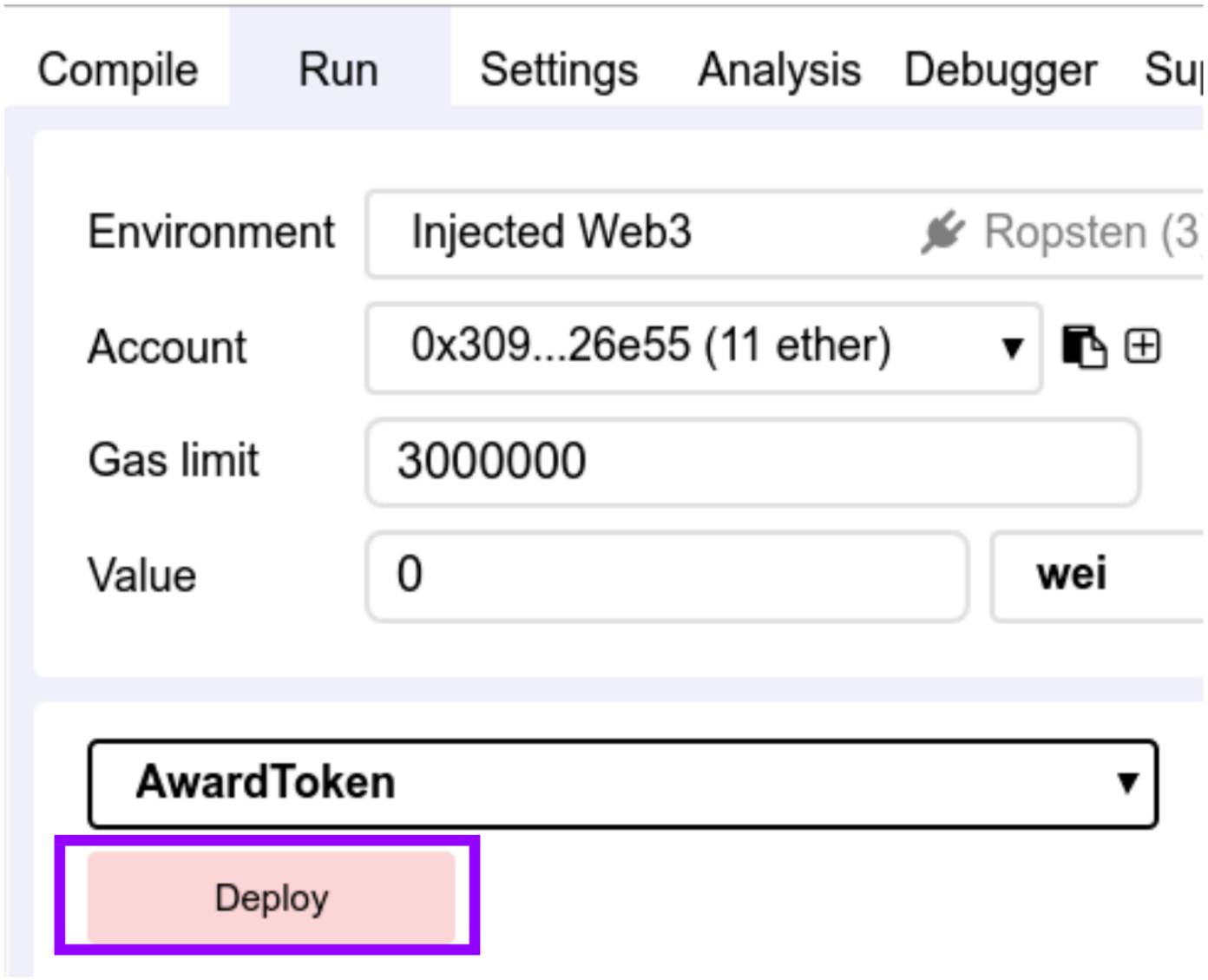


Imported Contracts



Deploy the contract

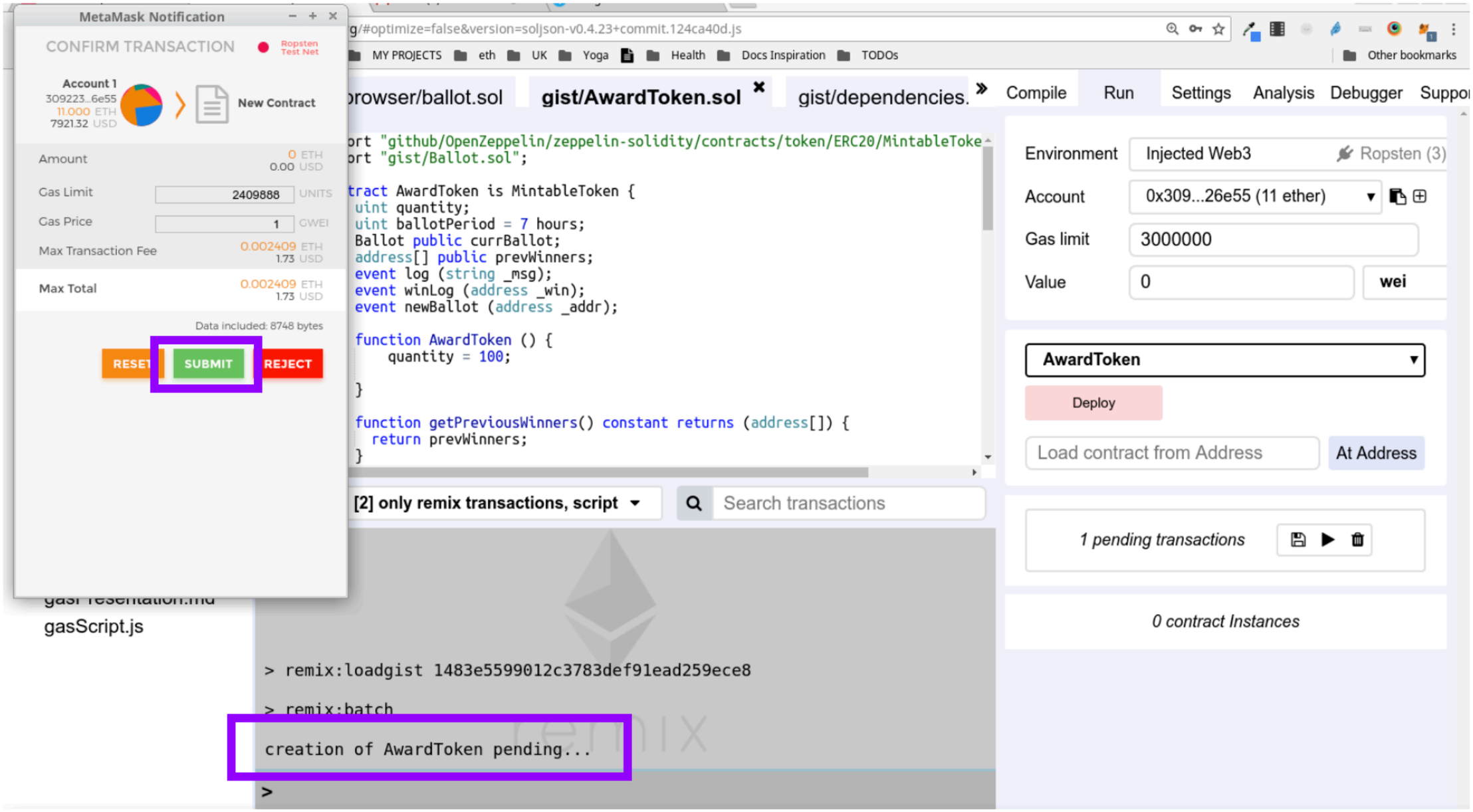
Run tab: Deploy button



Confirm the transaction

Submit button

But make sure you put in a gas price!



Check if tx is mined

Terminal logs in Remix

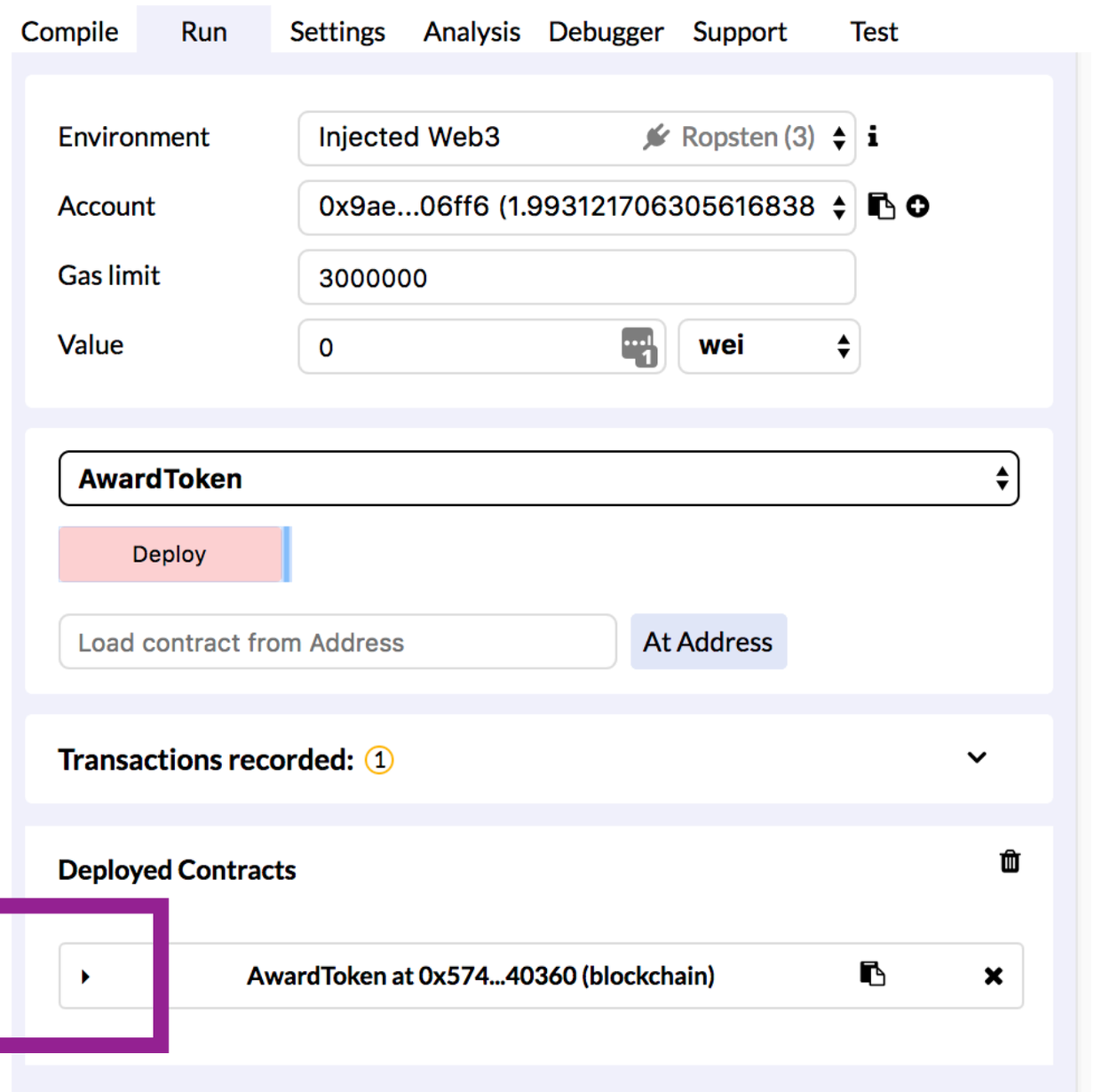
```
creation of AwardToken pending...

https://ropsten.etherscan.io/tx/0x404a4445ebb3a969b15257a586a61582afa07
dcf02b1b2617f77519b30378be8
```

```
▶ [block:3159099 txIndex:2]  from:0x309...26e55
to:AwardToken.(constructor) value:0 wei data:0x608...70029
logs:0 hash:0x404...78be8
```

Debug

Click to see the contract's UI

On the deployed contract

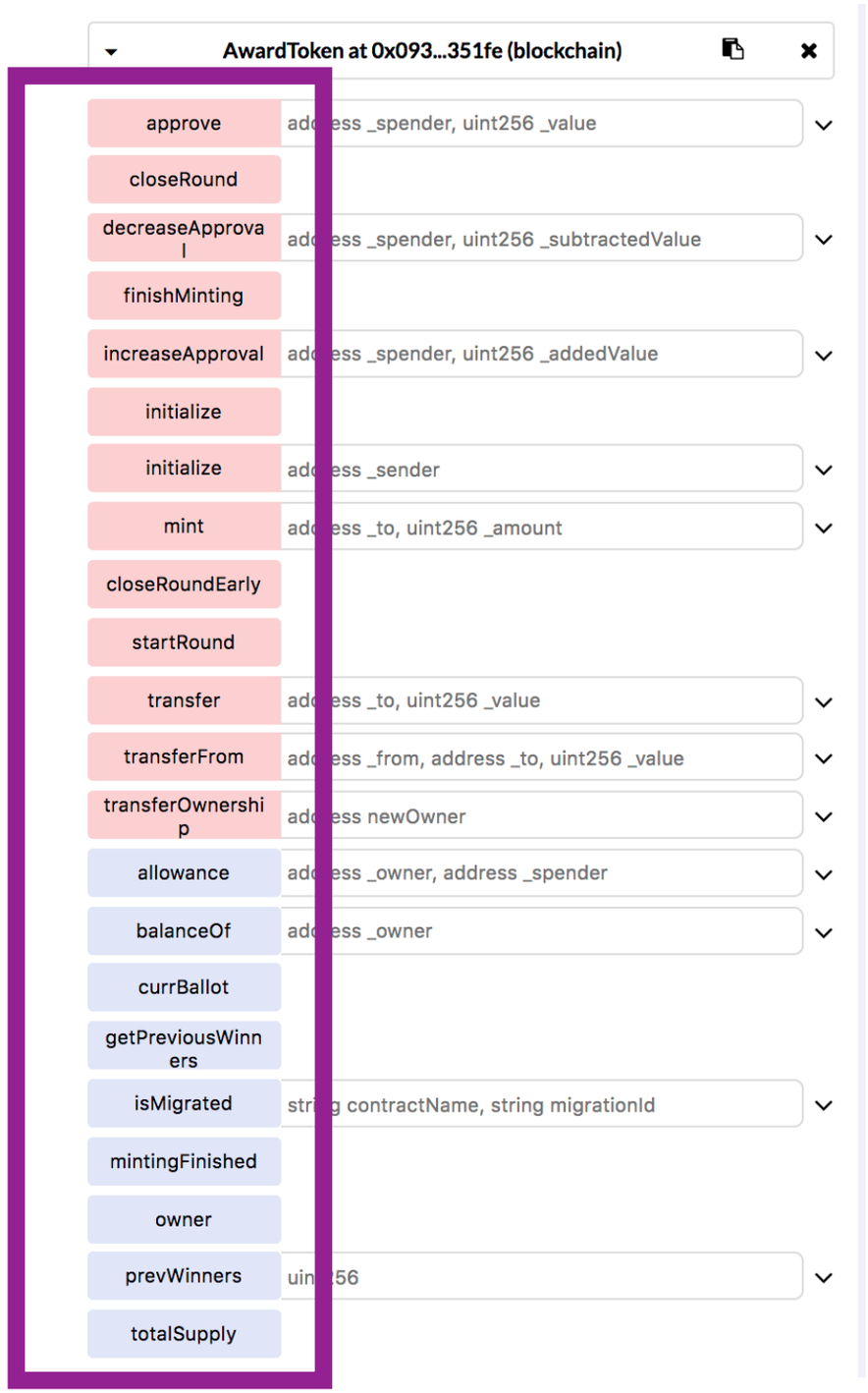


Voilà!

The Interactive UI for AwardToken.sol contract

Update image

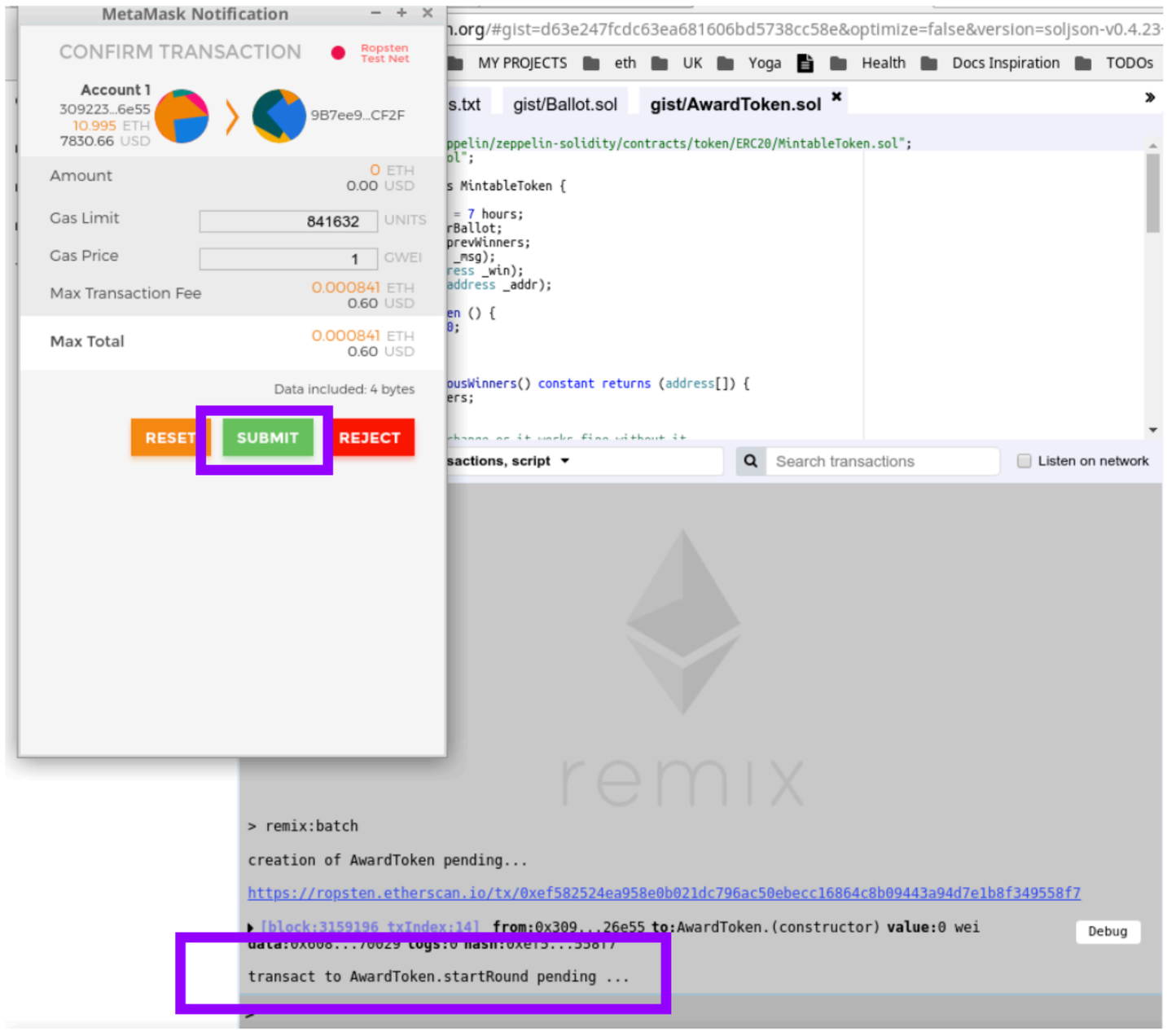
These are all the methods of AwardToken and the classes that it imported.



Execute startRound

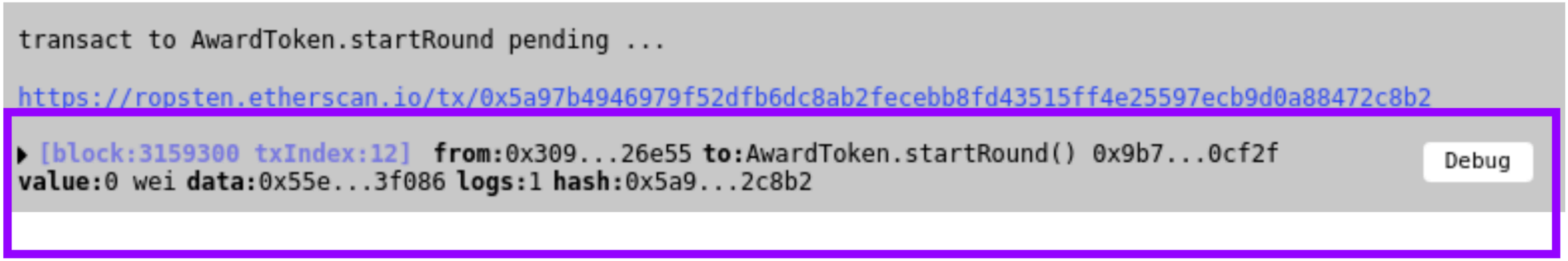
Its a payable function
(as opposed to a call function - which is free)

Confirm the transaction



Check if tx is mined

In the terminal logs in Remix



Expand tx log

to see the logs

[block:3665523 txIndex:4] **from:**0x9ae...06ff6 **to:**AwardToken.startRound() 0x574...40360 **value:**0 wei **data:**0x55e...3f086 **logs:**1 **hash:**0x16c...0a81c

Debug

status	0x1 Transaction mined and execution succeed
transaction hash	0x16c8af5a3fd0e5bcacd8858ab42d4f8eff39fc33bb98290740c03eeb4880a81c
from	0x9ae59af2e33480caa48f2dc6f6cede7ffab06ff6
to	AwardToken.startRound() 0x574d270dc04e89c5d65e24e19f1deb9e17240360
gas	613643 gas
transaction cost	613643 gas
hash	0x16c8af5a3fd0e5bcacd8858ab42d4f8eff39fc33bb98290740c03eeb4880a81c
input	0x55e...3f086
decoded input	{}
decoded output	-
logs	[{ "from": "0x574d270dc04e89c5d65e24e19f1deb9e17240360", "topic": "0x65f35fb257c91daed794331bfd2ad0f4439d49319d52a5b3bfb04c84969fdbeb", "event": "newBallot", "args": { "0": "0xD6052C85A3D26eE9EeC8262d462bfDC672B80D93", "_addr": "0xD6052C85A3D26eE9EeC8262d462bfDC672B80D93", "length": 1 } }]
value	0 wei

Checkout the startRound function in the editor

gist/dependencies.js | **gist/AwardToken.sol**

ContractDefinition **AwardToken** 0 reference(s)

```
23    function startRound() onlyOwner canMint public returns (bool) {
24        // if this is the first minting then we should let this go immediately
25        if (address(currBallot) == 0x0) {
26            currBallot = new Ballot(ballotPeriod);
27            newBallot(currBallot);
28        } else {
29            revert();
30        }
31    }

33    function closeRoundEarly () onlyOwner {
34        if (address(currBallot) != 0x0 && !currBallot.timeOut()) {
35          currBallot.finish();
36        } else revert();
37    }
38
39    function closeRound() onlyOwner {
40        //  this can only be done by the owner of the contract
41
42        if (address(currBallot) != 0x0 && currBallot.timeOut()) {
43            // get winner
44            address winner = currBallot.winningProposal();
45            winLog(winner);
46            // send to winner - but first make sure the address is valid
47            if ( winner == 0x0){
48                log("no winner");
49            } else {
```

Get ballot's address

Execute currBallot call

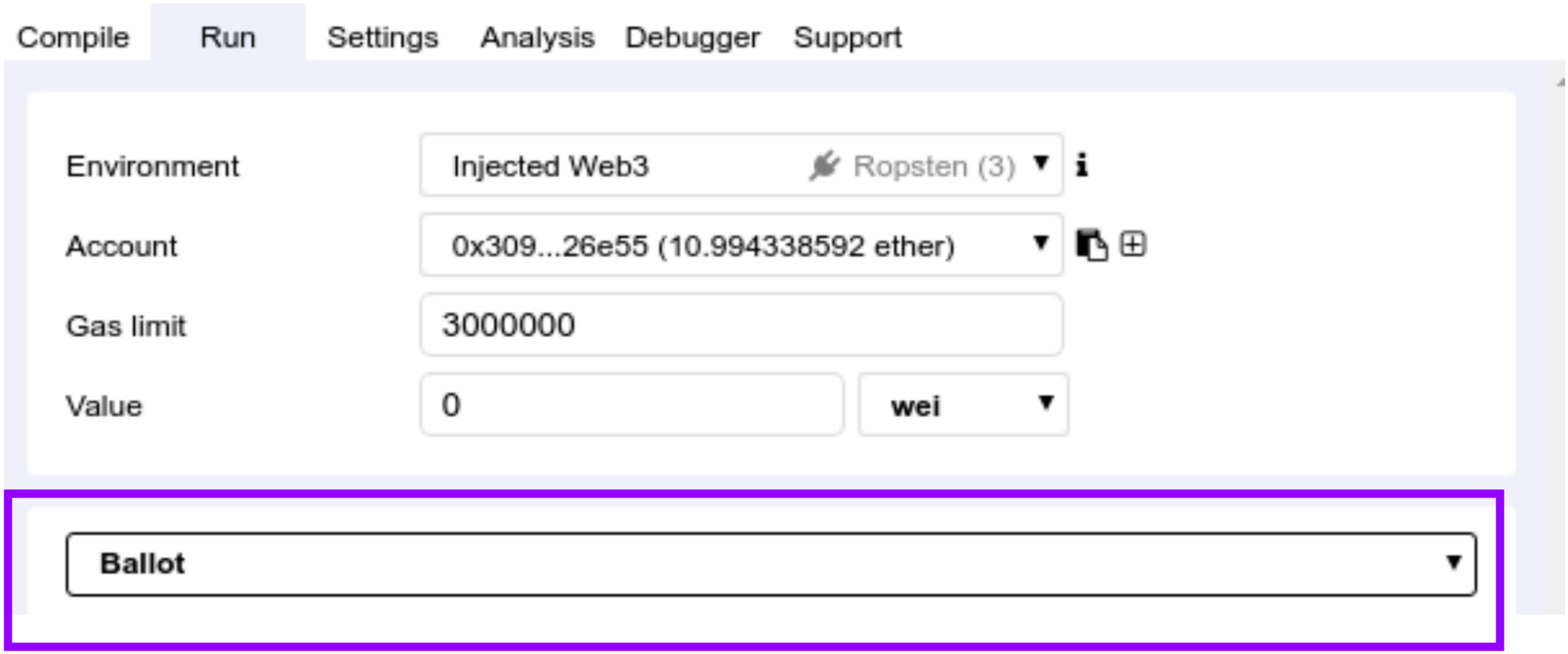
Copy ballot's address

currBallot output

Switch to Ballot

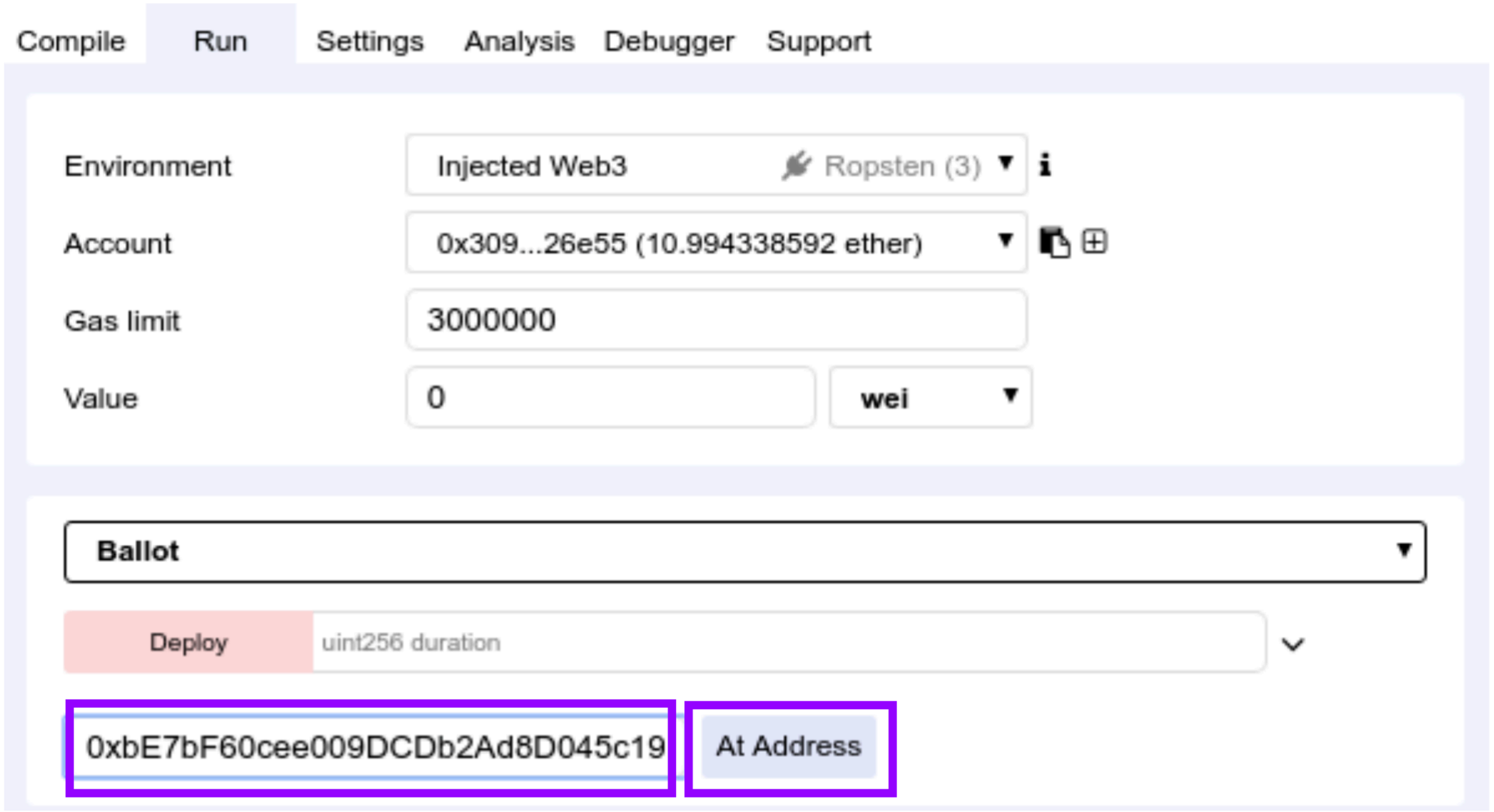
(which is loaded from AwardToken.sol)

Run tab: dropdown



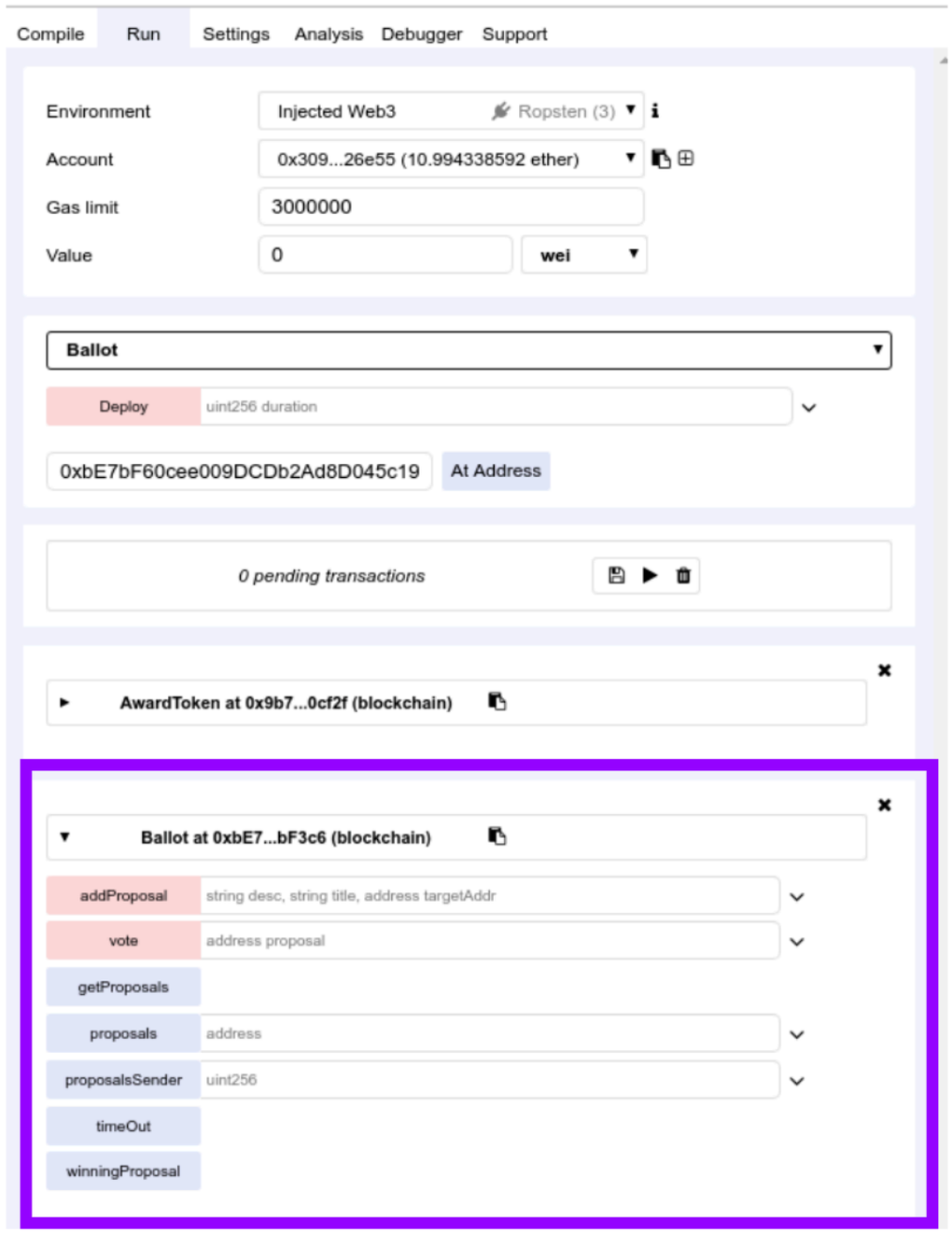
Access Ballot contract

Paste address + click At Address



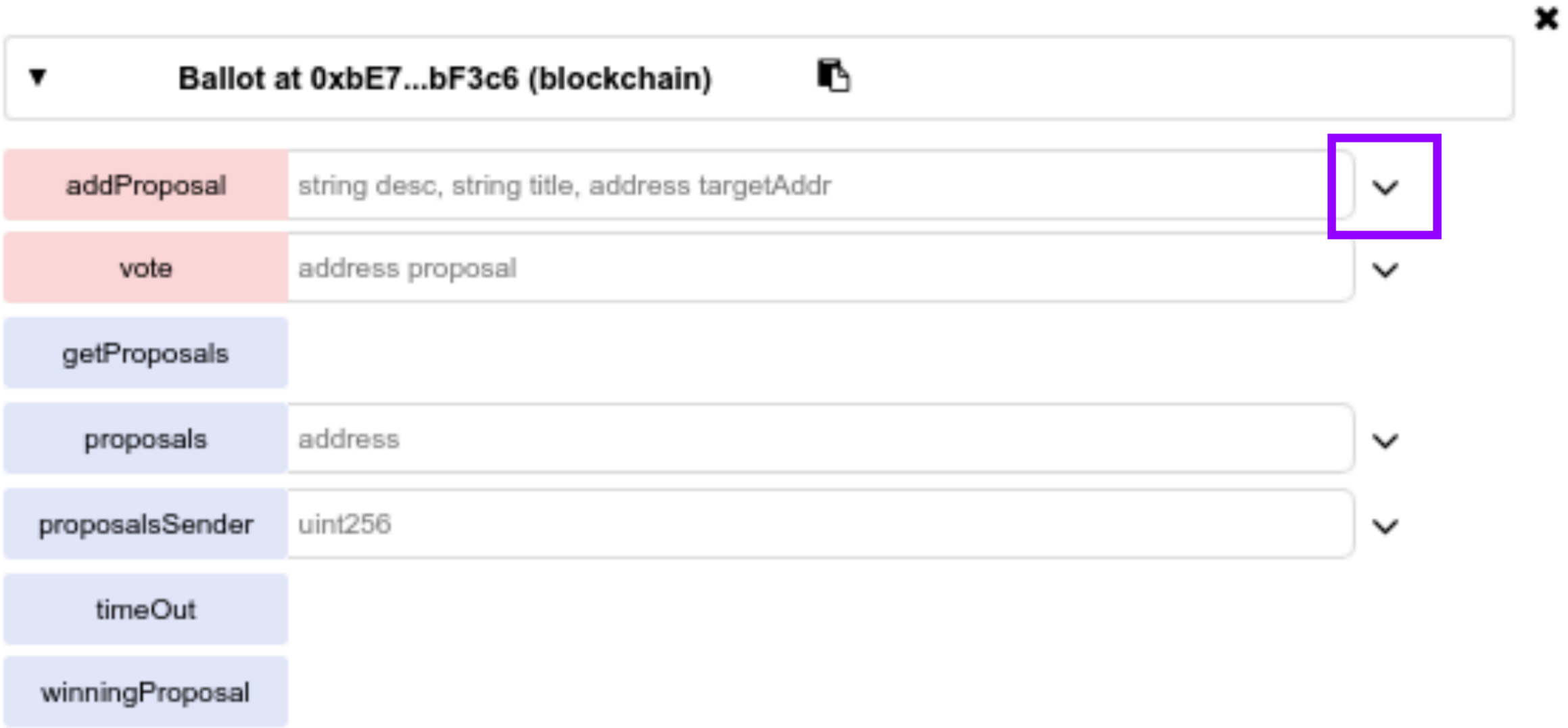
See autogenerated UI

Interactive UI for Ballot.sol contract



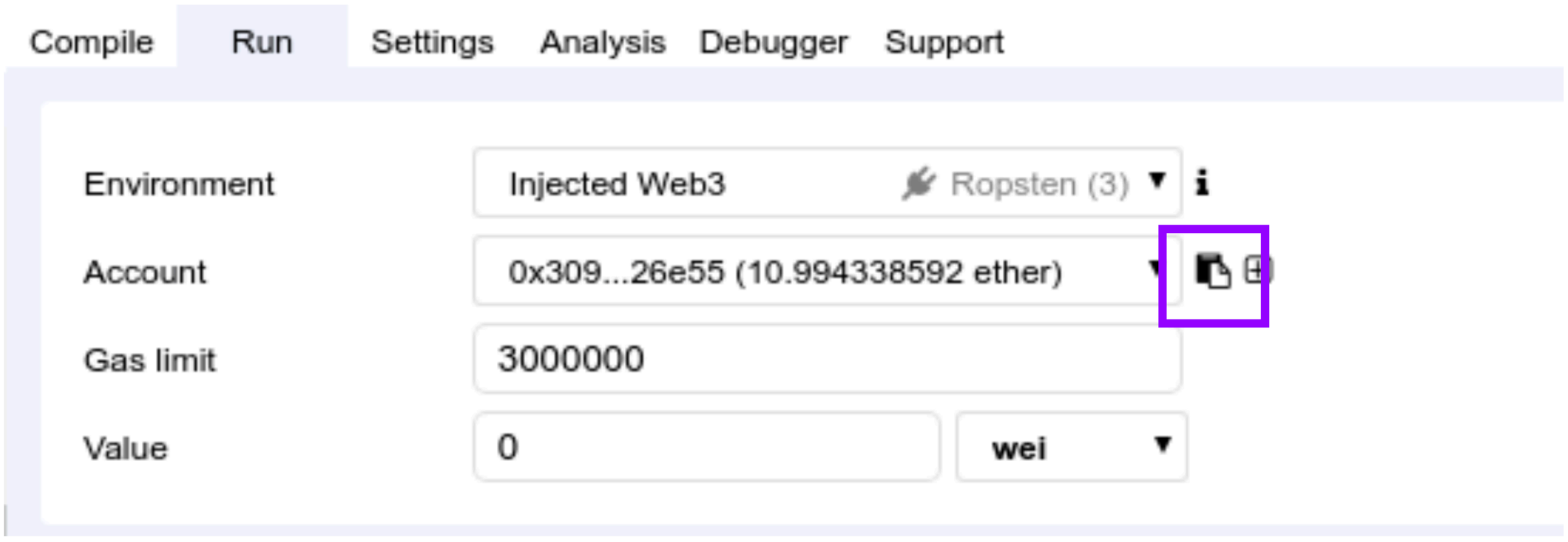
Add a new proposal

Expand addProposal function



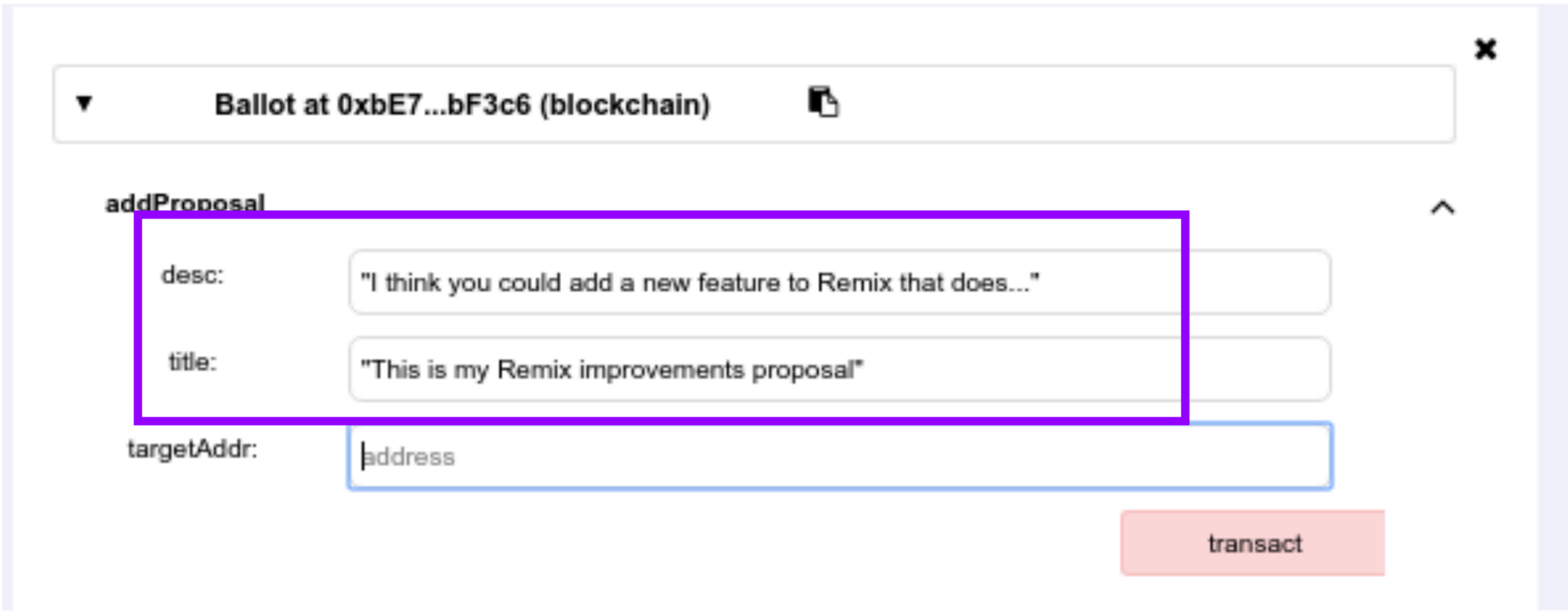
Copy your address

Run tab: Account



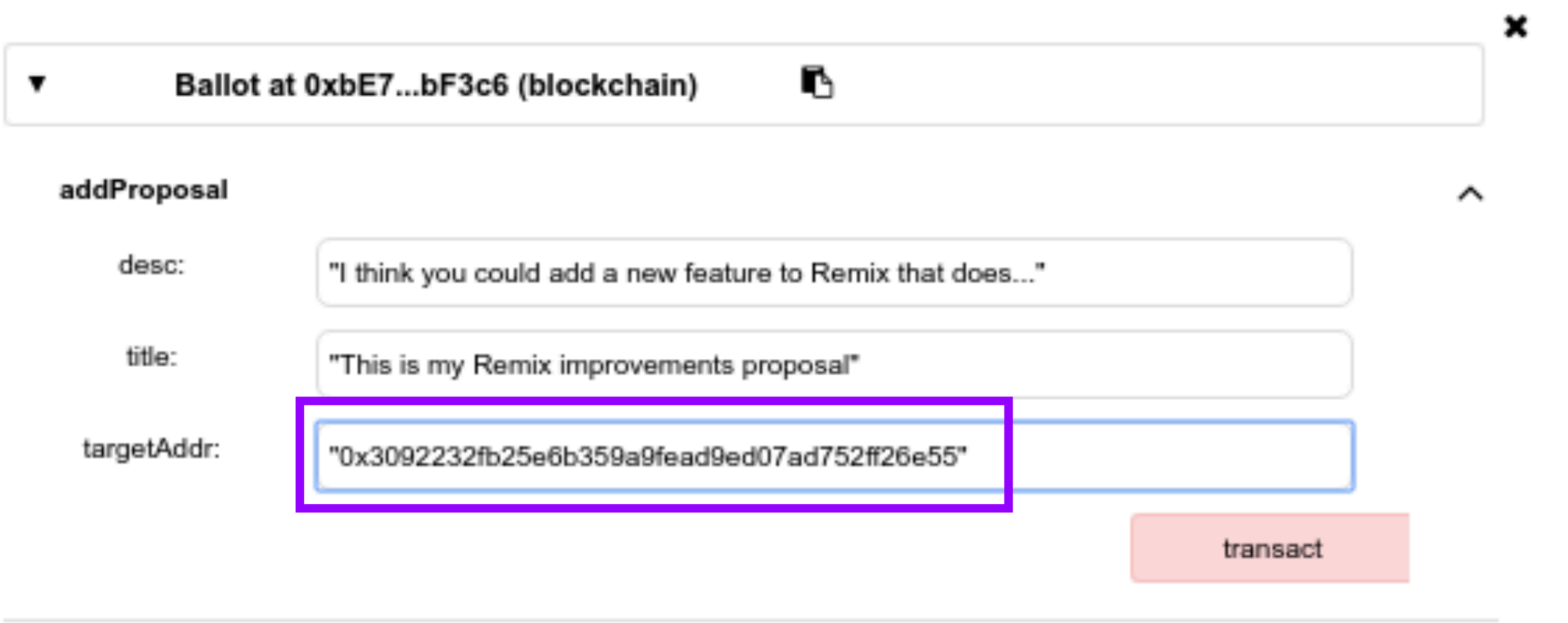
Type a proposal

Run tab: Account



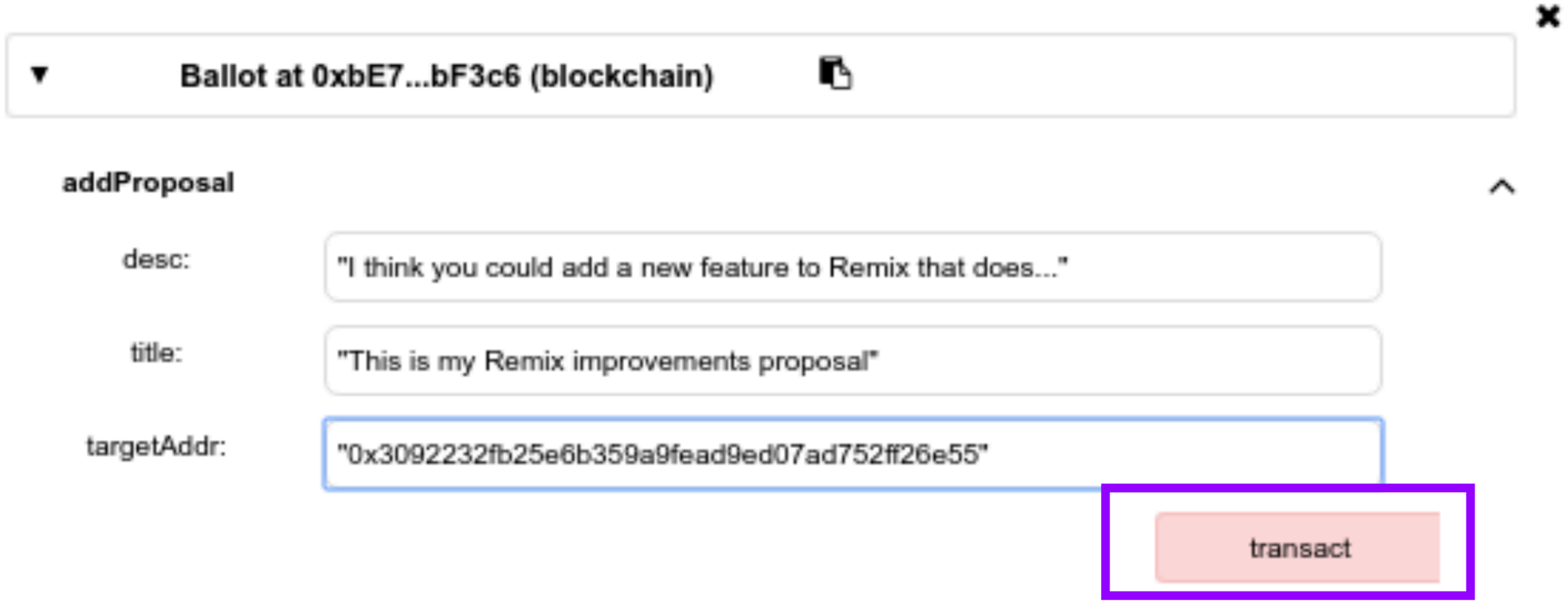
Add your address

Paste the address



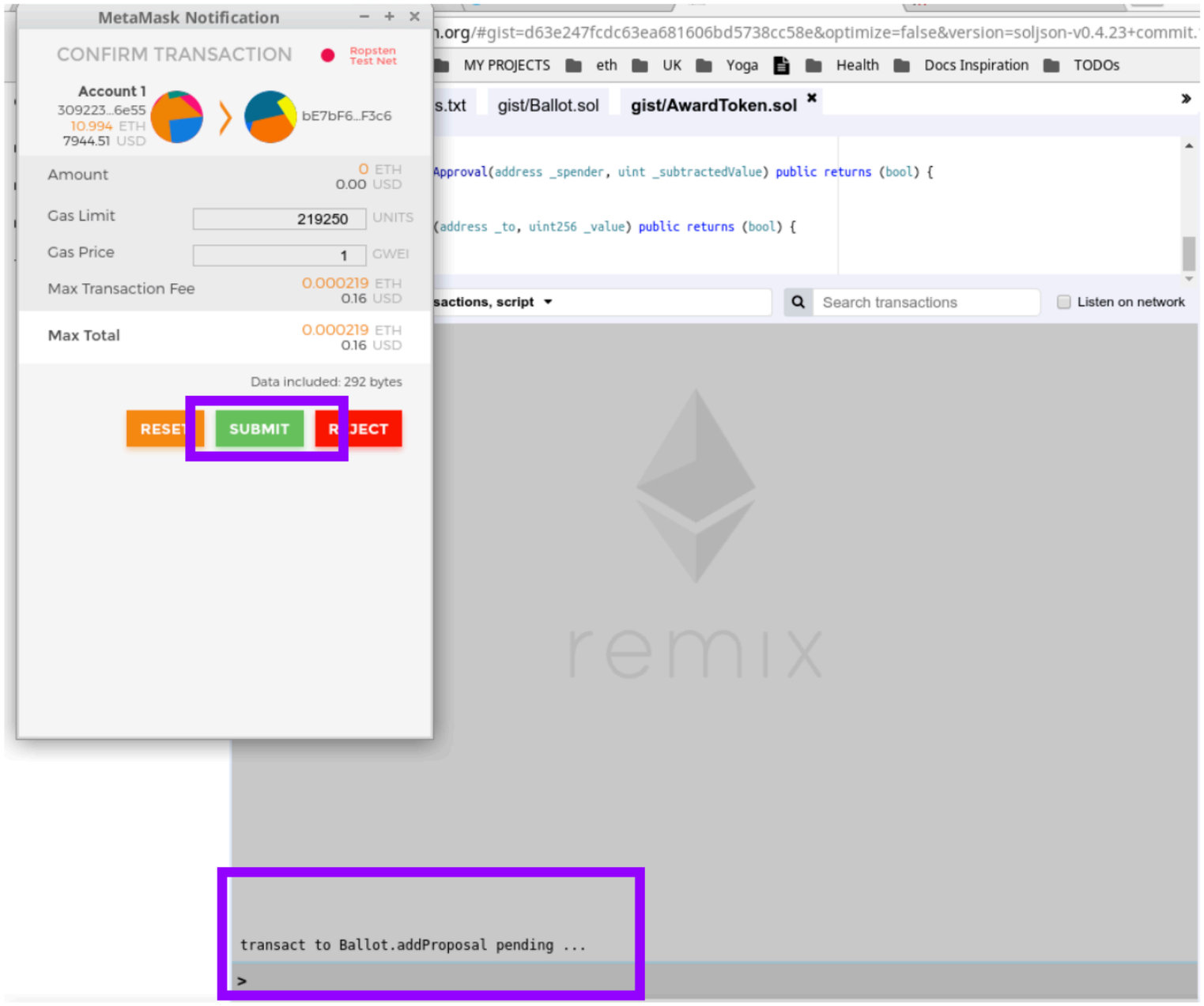
Execute addProposal

transact button



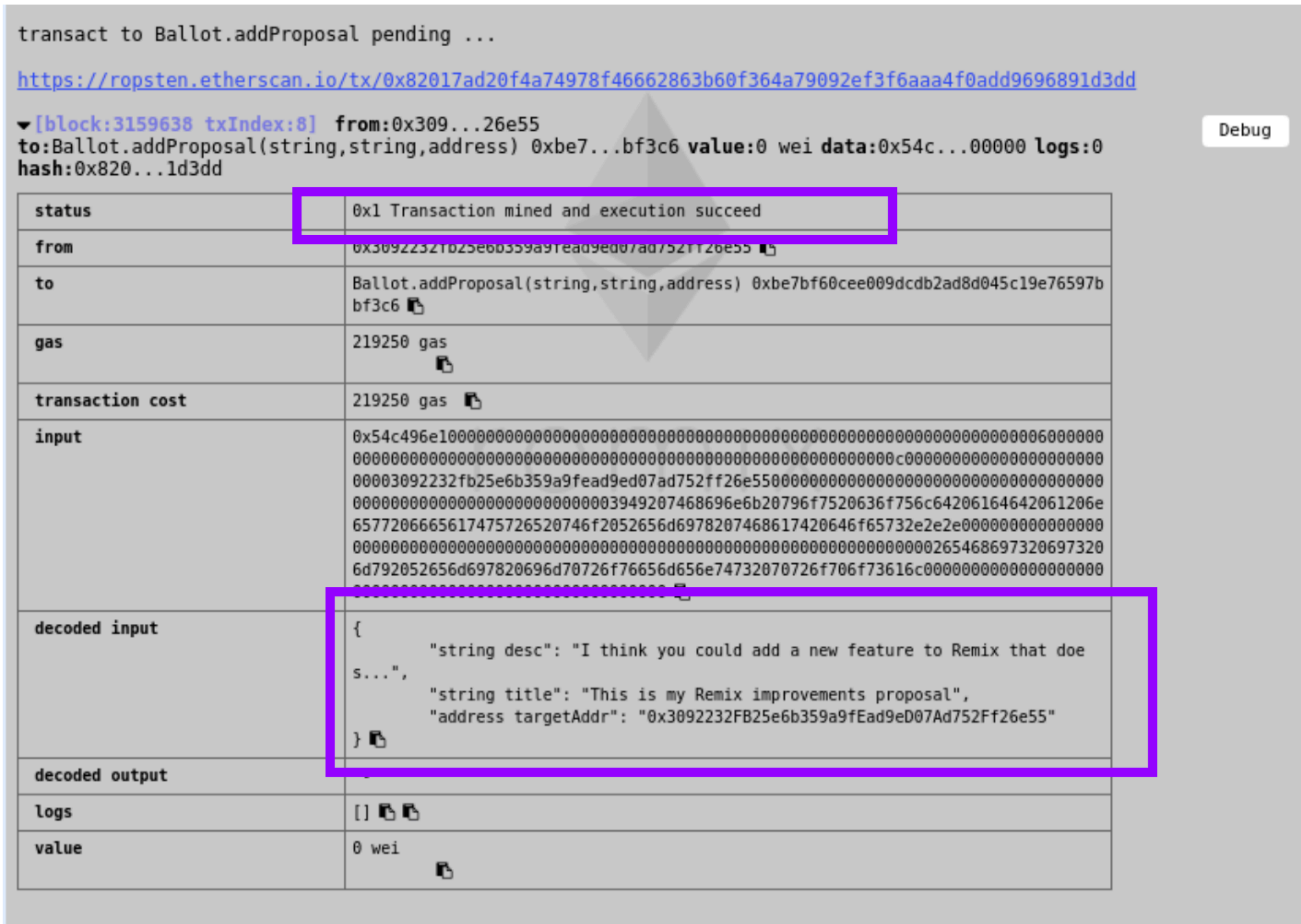
Confirm the transaction

Submit button



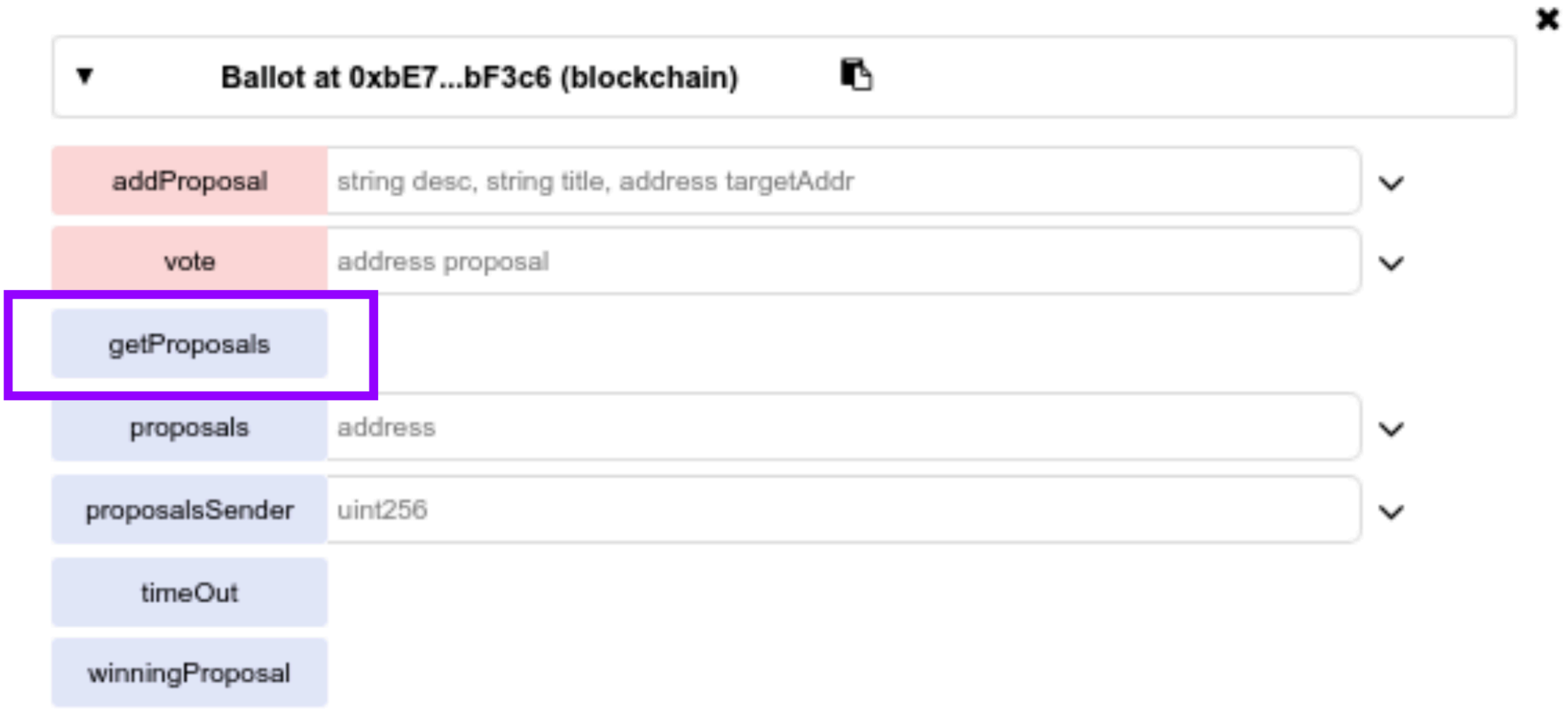
Check if tx succeeded

Terminal logs in Remix



Execute getProposals

getProposals call



try it live!

See Proposals Addresses

well in so far there will only be 1 address

call to Ballot.getProposals

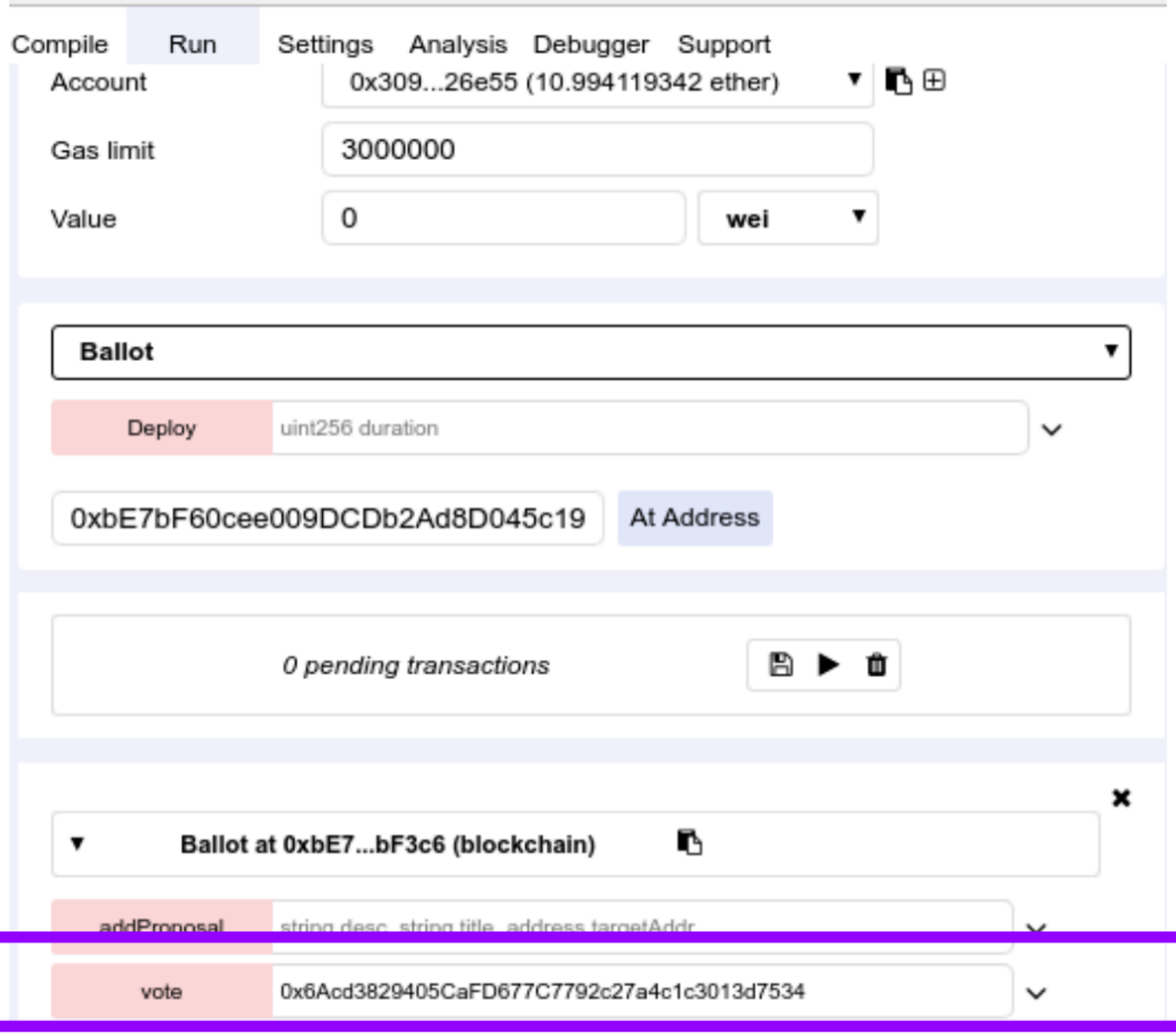
▼[call] **from:**0x3092232fb25e6b359a9fead9ed07ad752ff26e55 **to:**Ballot.getProposals()
data:0x625...64c48

Debug

from	0x3092232fb25e6b359a9fead9ed07ad752ff26e55
to	Ballot.getProposals() 0xbE7bF60cee009DCDb2Ad8D045c19e76597bbF3c6
input	0x62564c48
decoded input	{}
decoded output	{ "0": "address[]: 0x3092232FB25e6b359a9fEad9eD07Ad752Ff26e55,0xFd0f51afb685Cd8735AfE7685D21355589602b8c,0x6Acd3829405CaFD677C7792c27a4c1c3013d7534" }
logs	[]

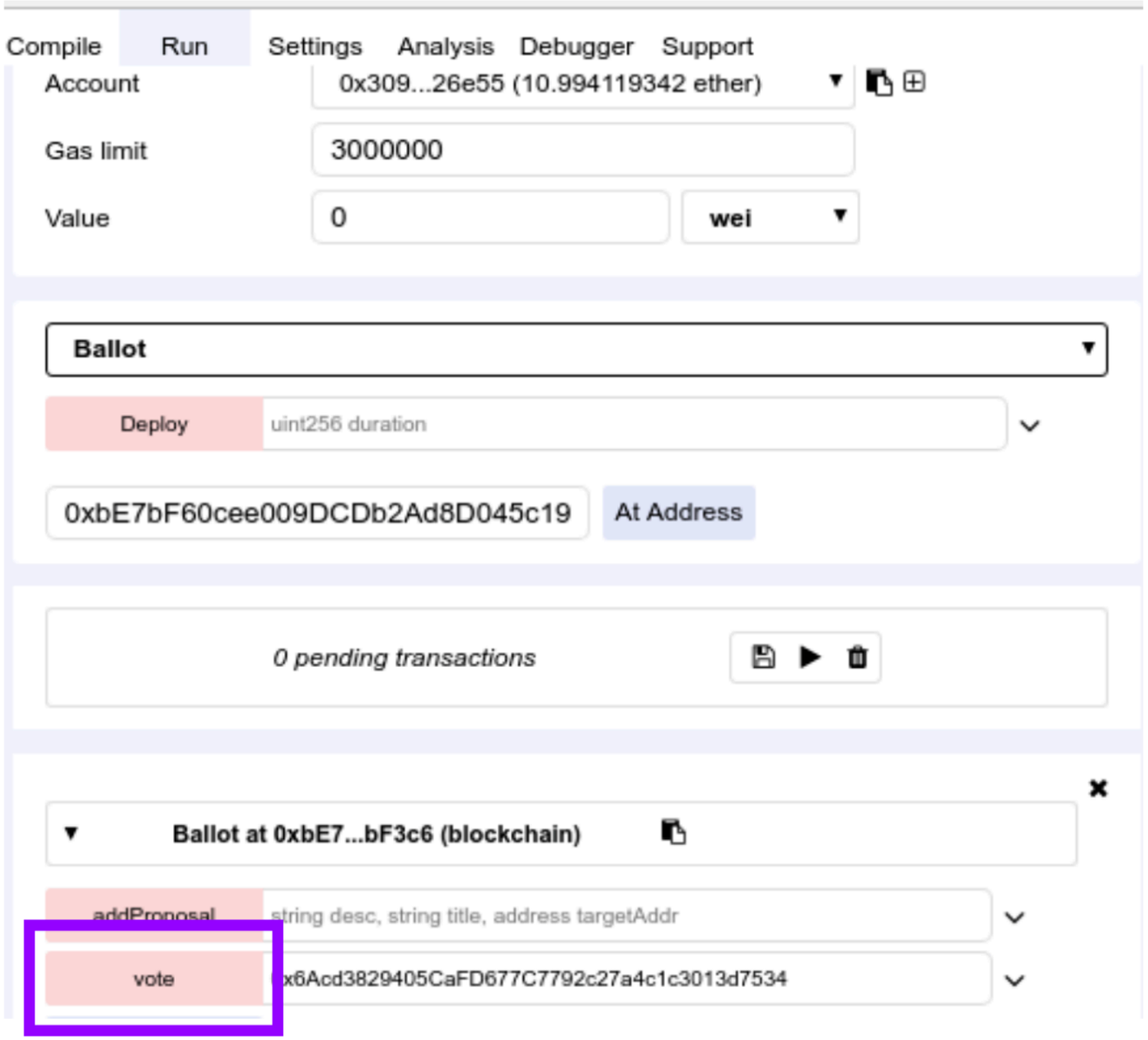
Vote for one Proposal

Paste Proposal Address you want to vote for



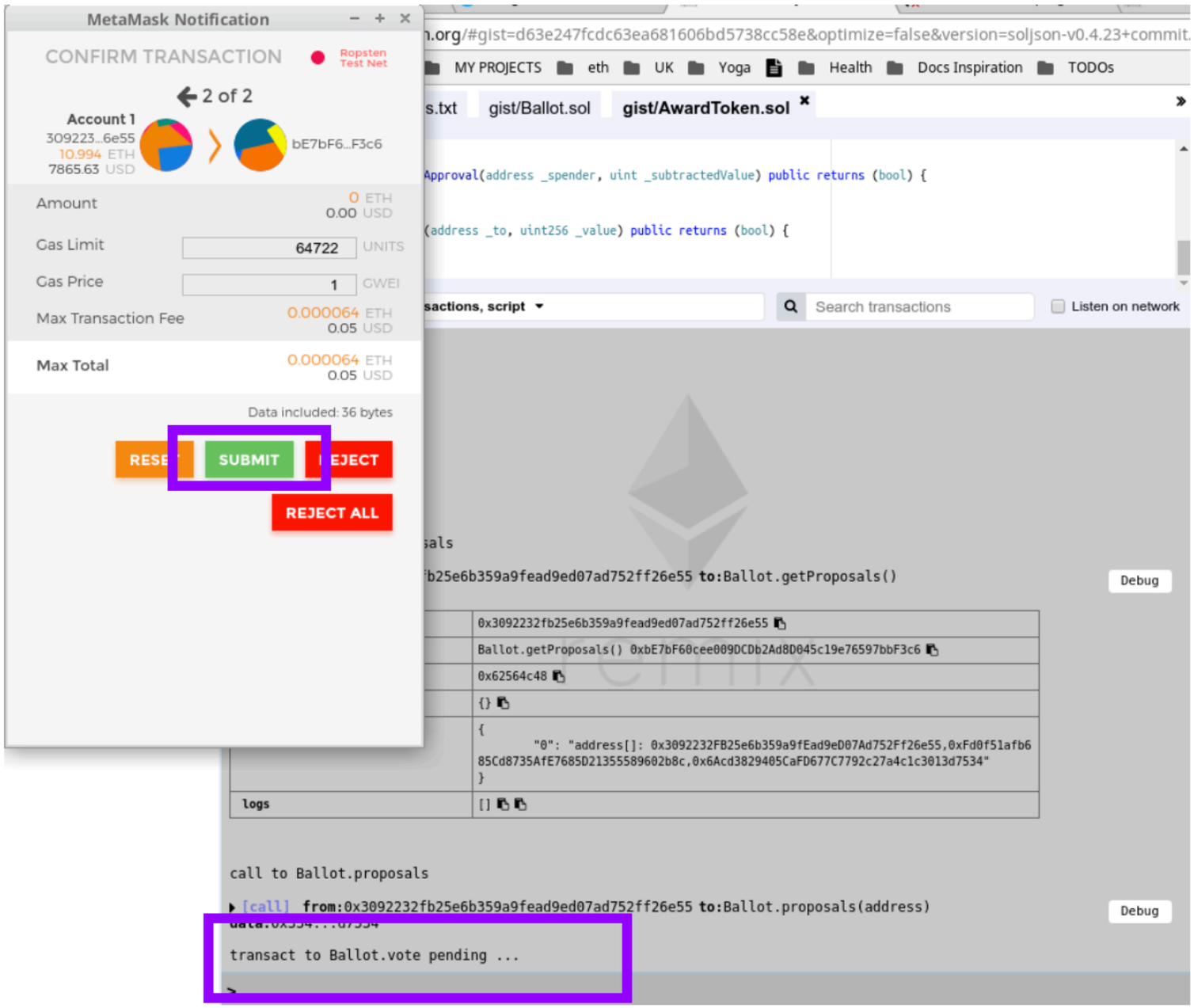
Execute vote transaction

vote button



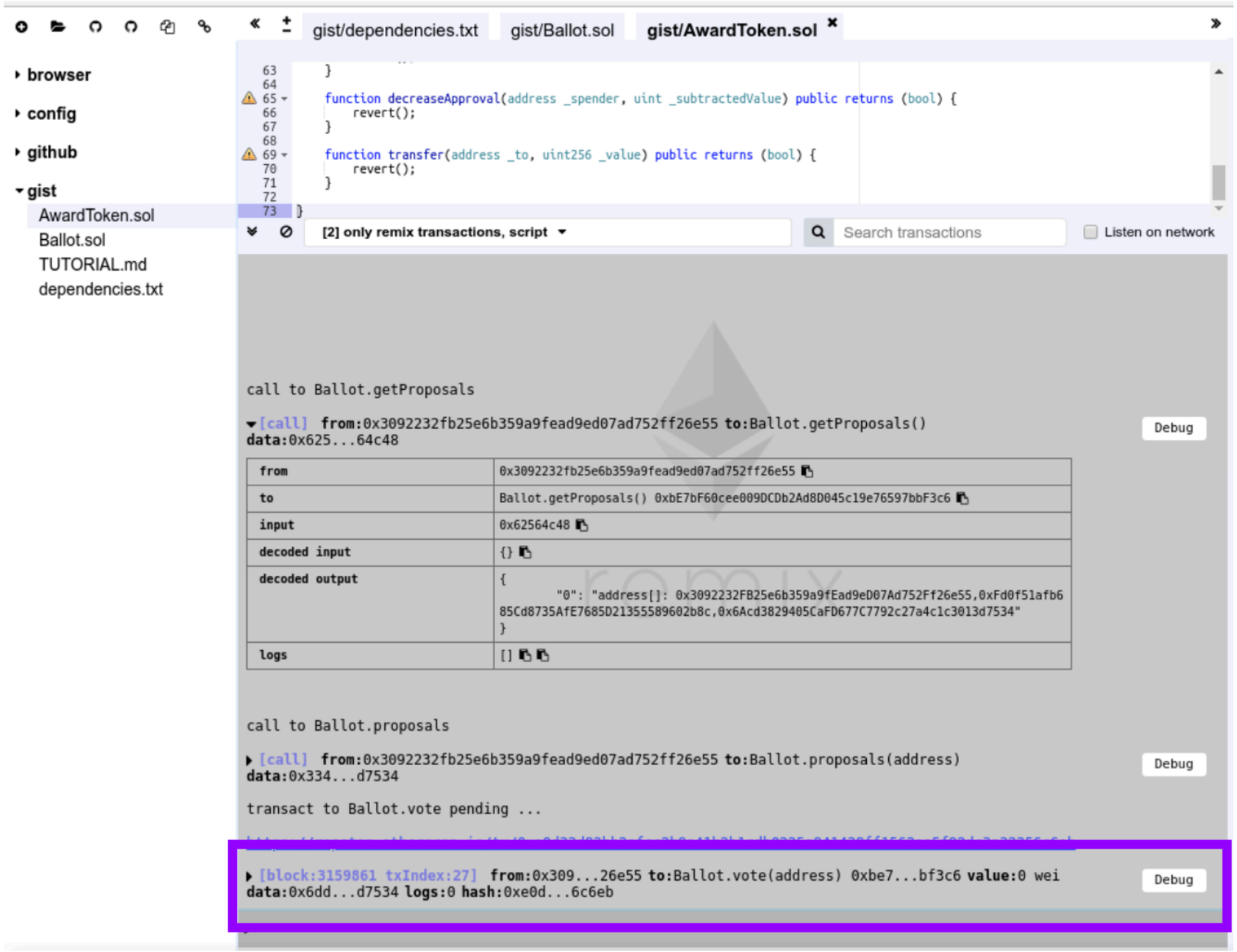
Confirm the transaction

Submit button



Check if tx succeeded

Terminal logs in Remix



Now let's try it out connecting a frontend

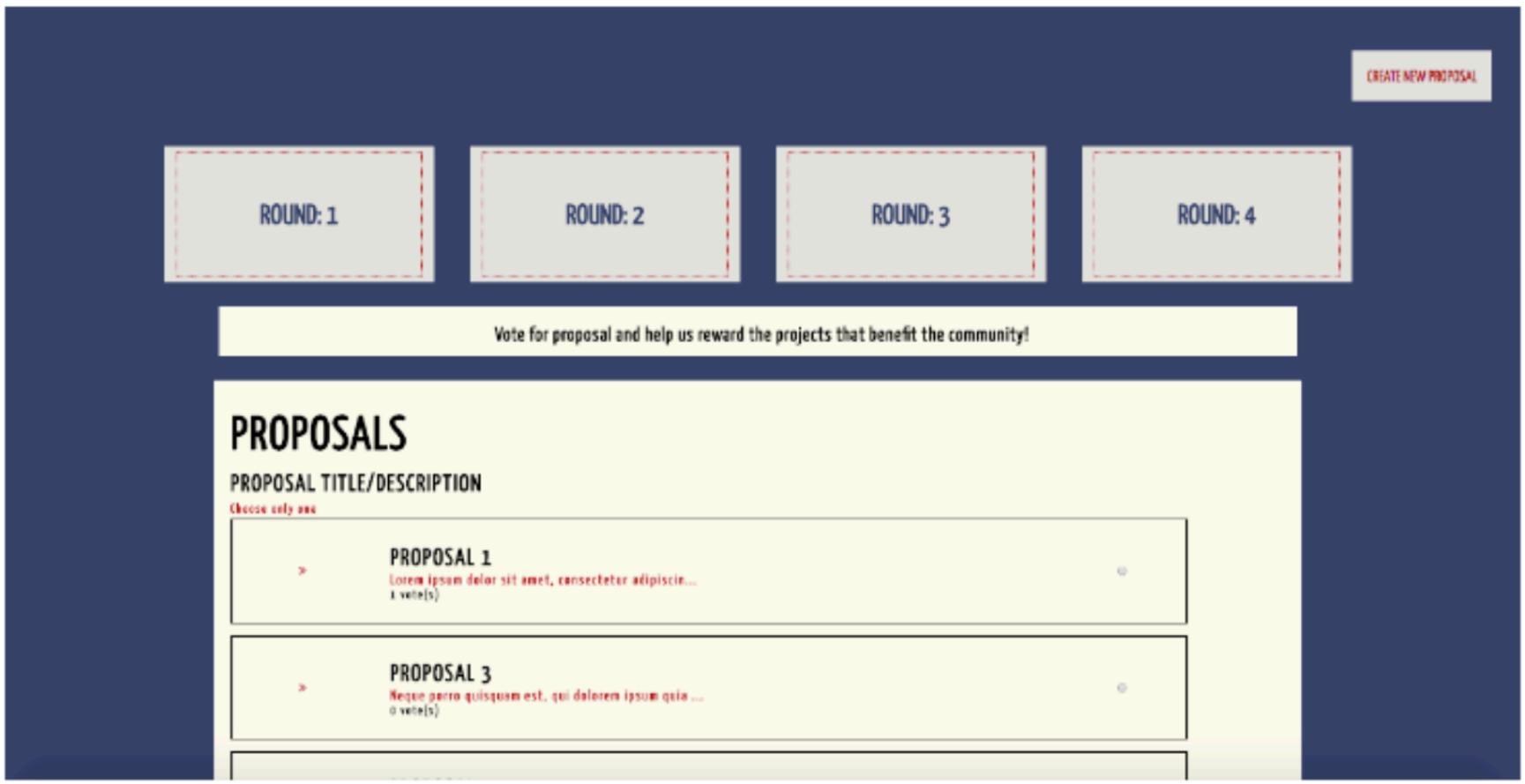
http://bit.ly/remix-voting

To access our Award Token from this frontend - you need the address of the Award Token.

Go to **ethereum/remix-workshop** to access the award token I just deployed

```
contract Ballot {

    uint _duration;
    uint _startTime;
    struct Proposal {
        string description;
        string title;
        uint voteCount;
```

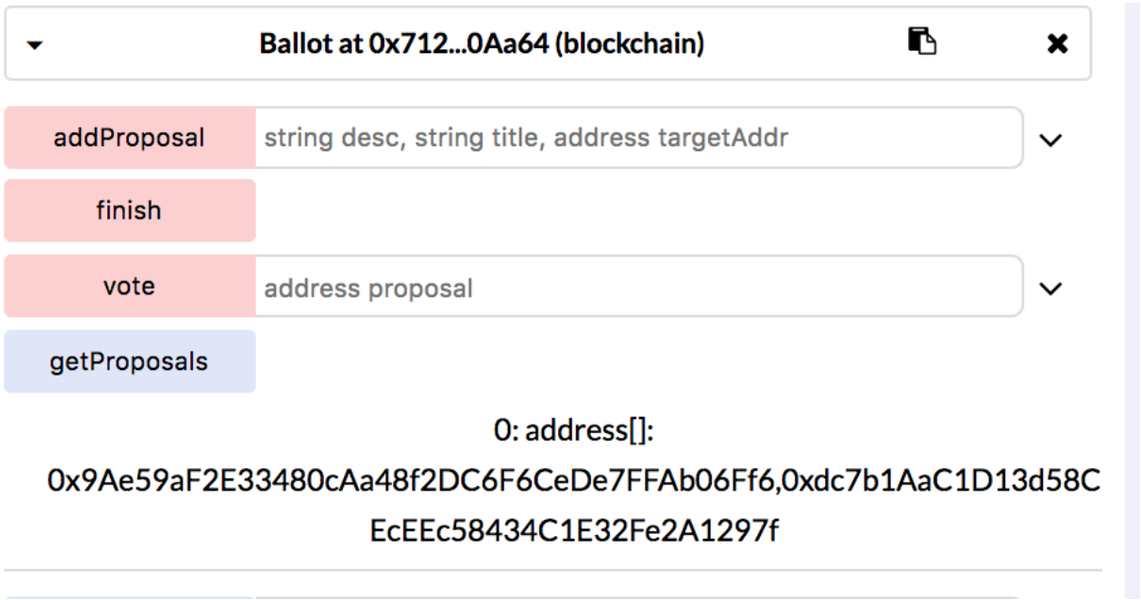
```
contract AwardToken is MintableToken {
    uint quantity;
    uint ballotPeriod = 7 hours;
    Ballot public currBallot;
    address[] public prevWinners;
```



Let's check results

http://bit.ly/remix-voting

Check the state of the contract



@ninabreznik @ryestew @yann300 @serapath @iurimatias

http://bit.ly/remix-workshop-repository