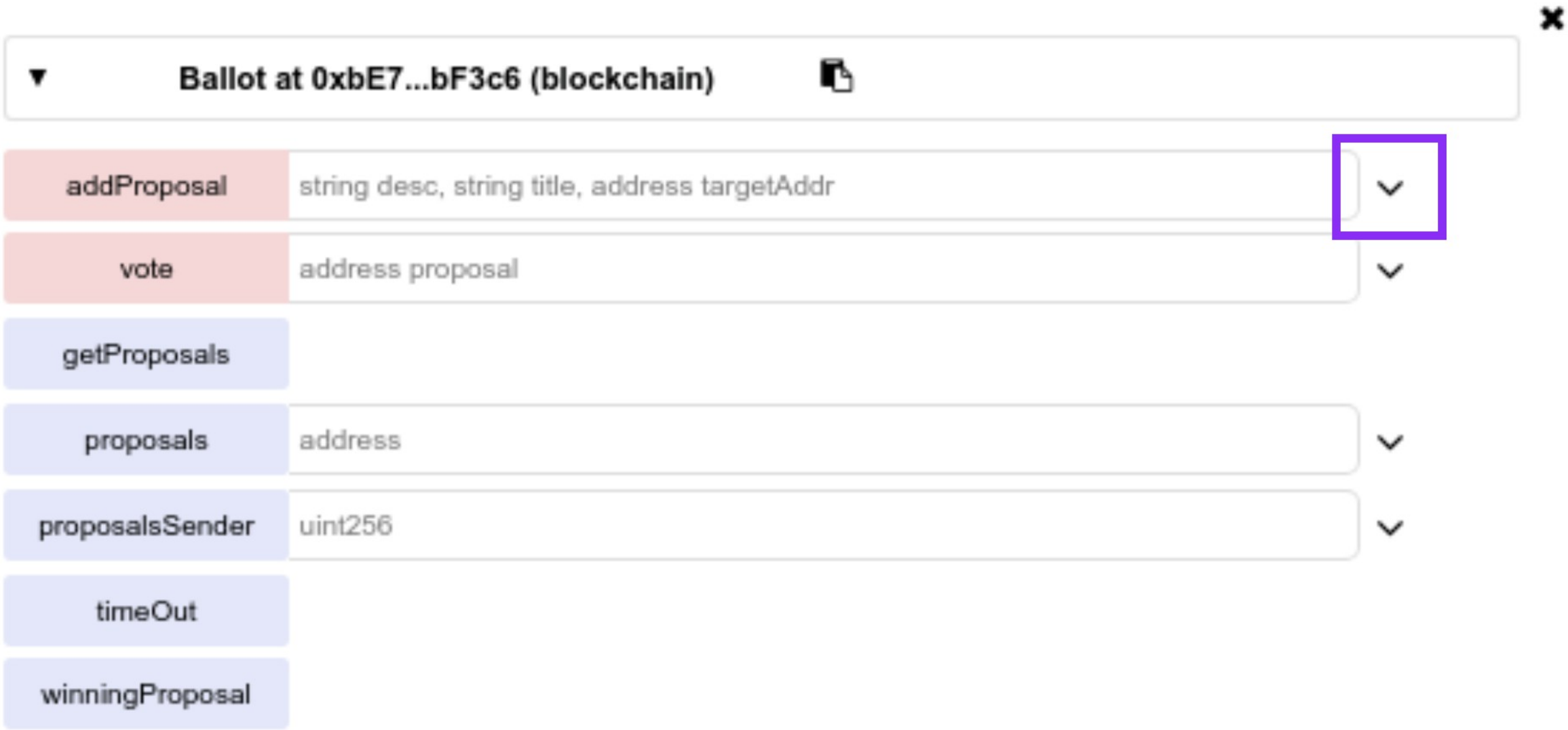
Part III

Adding a Proposal
&
Viewing the DAPP

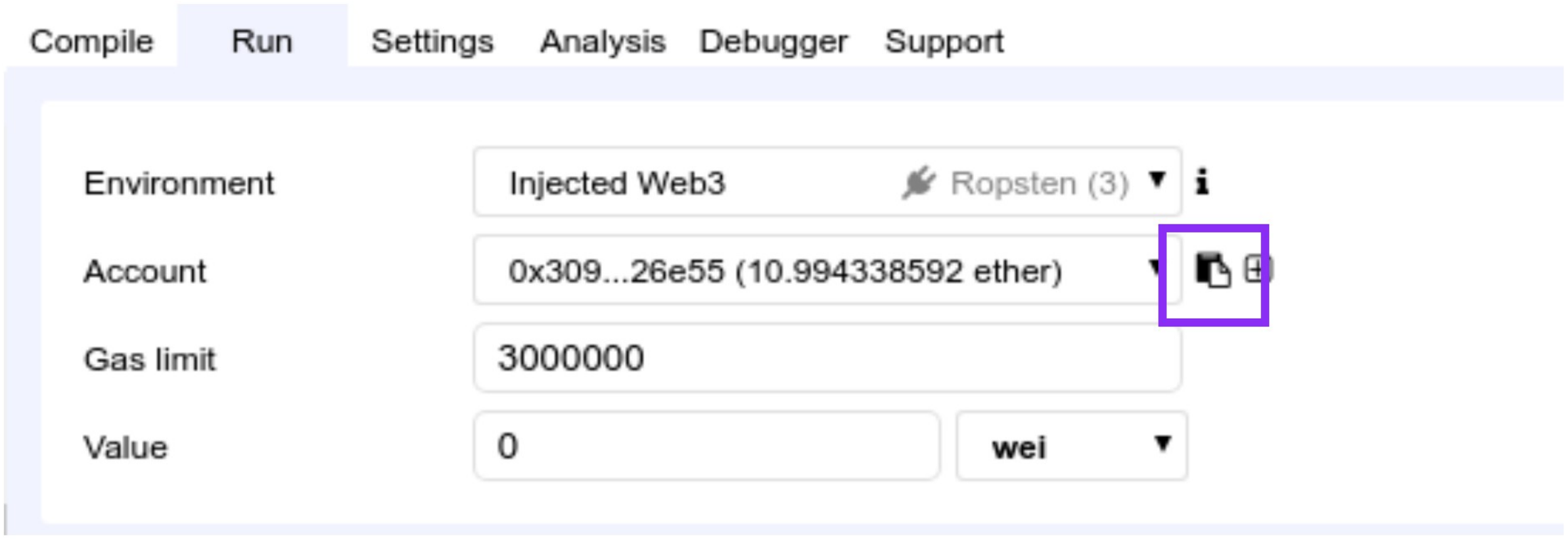
Add a new proposal

Expand addProposal function



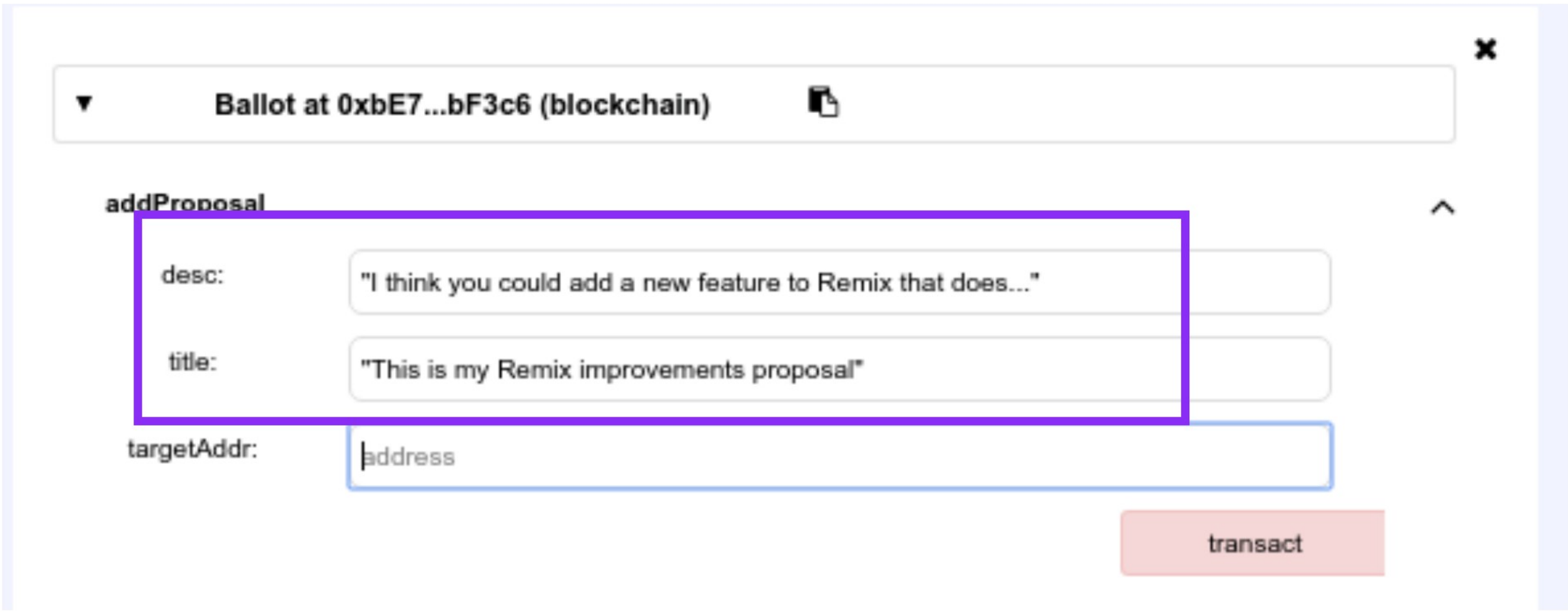
Copy your address

Run tab: Account



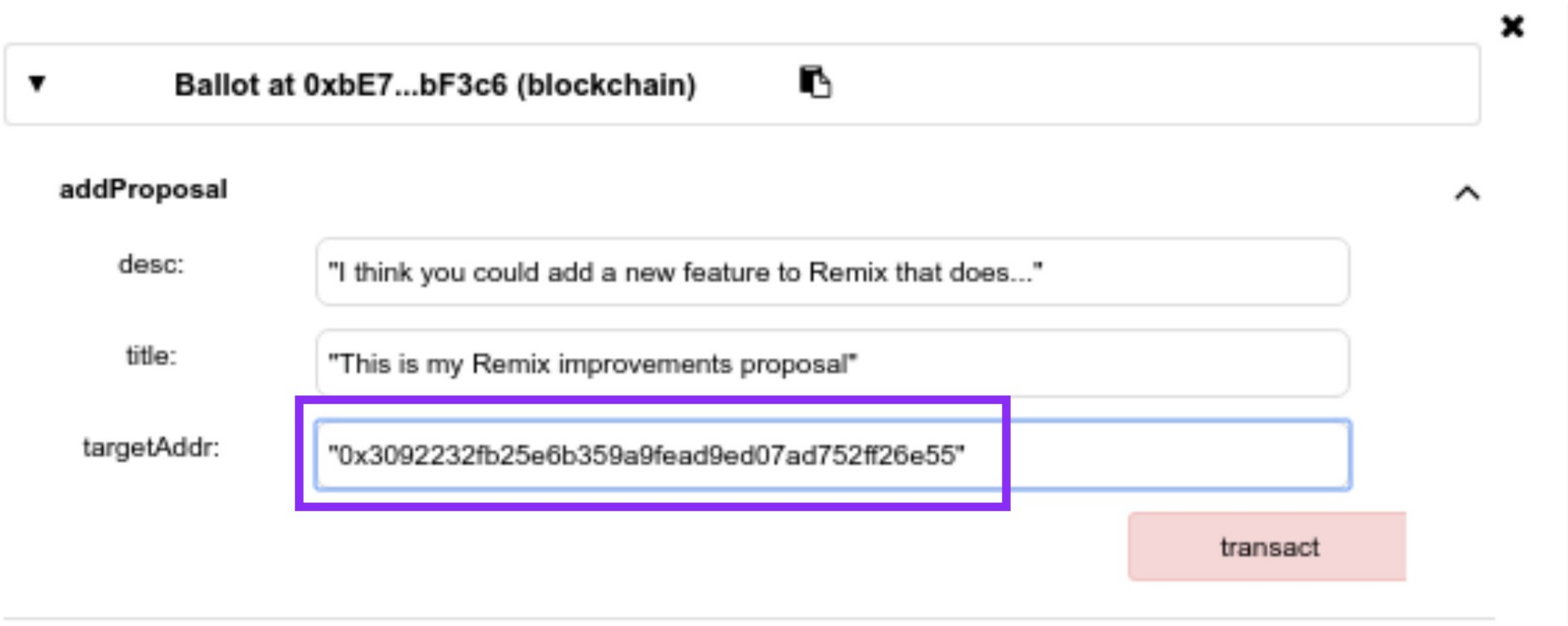
Type a proposal

Run tab: Account



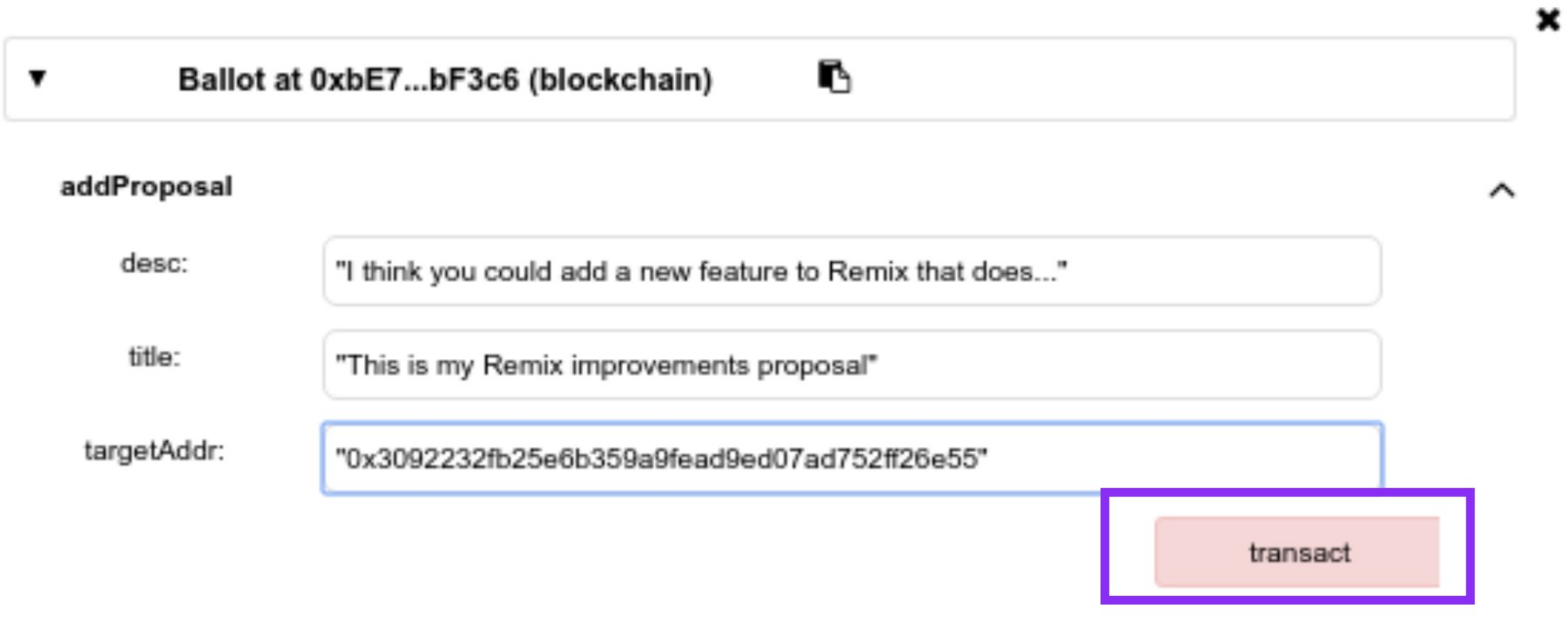
Add your address

Paste the address



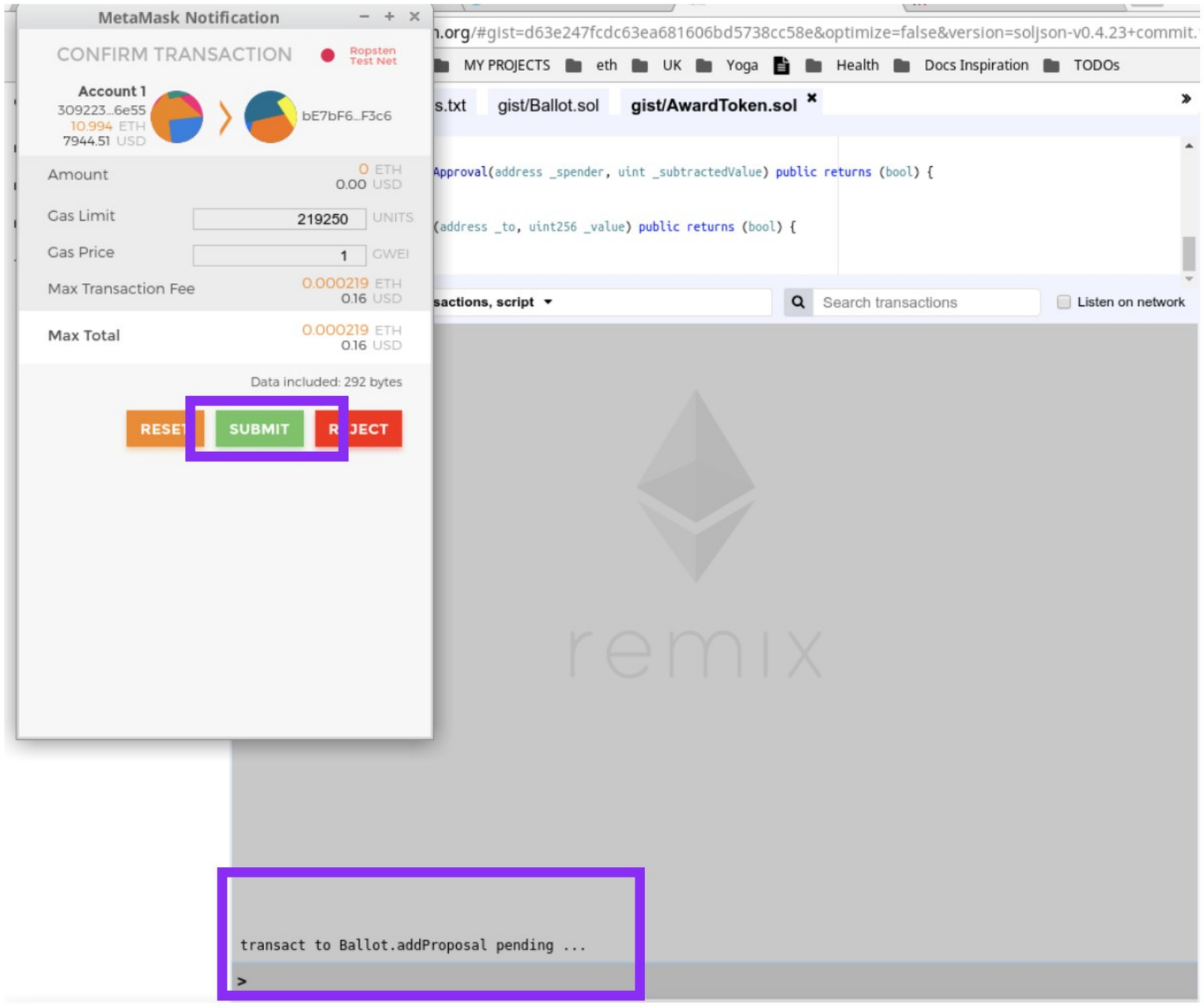
Execute addProposal

Hit the transact button



Confirm the transaction

Submit button

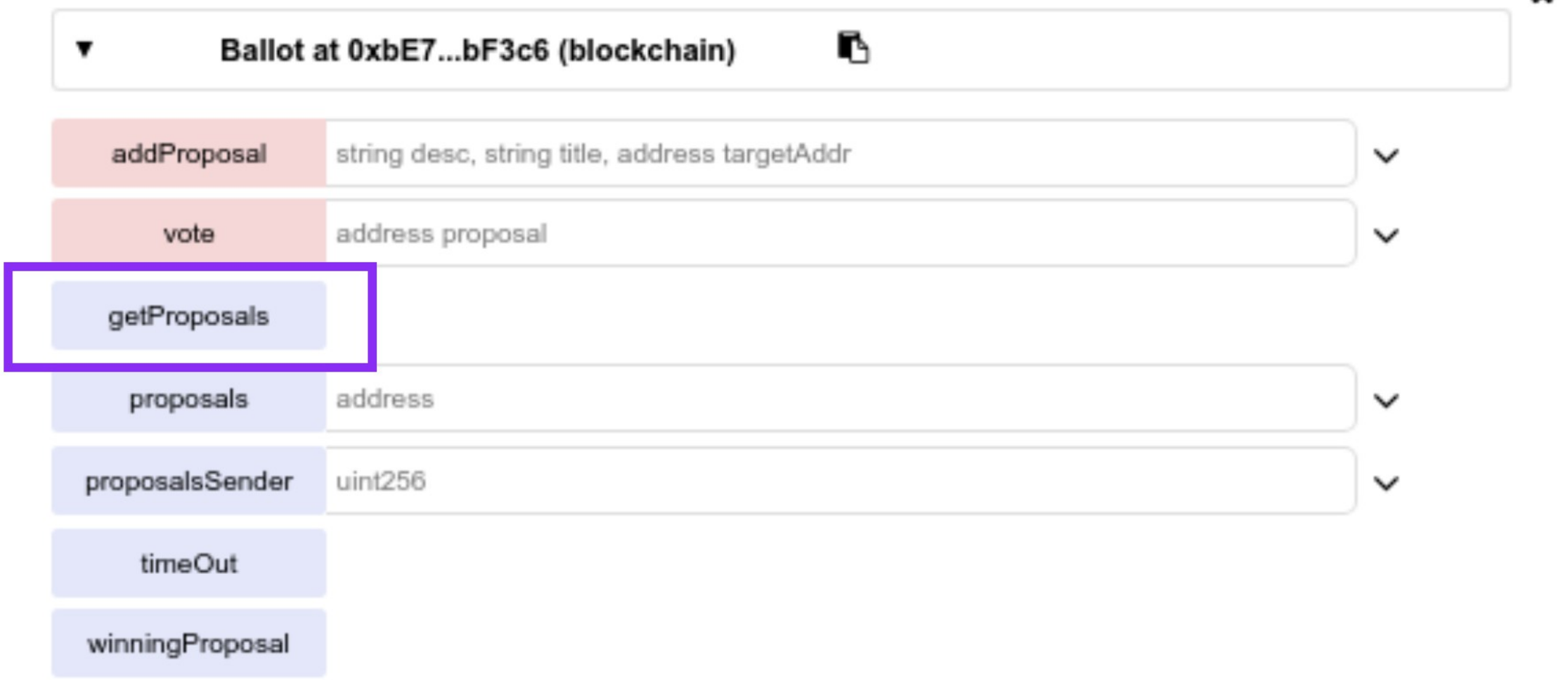


Check if tx succeeded

Terminal logs in Remix

Execute getProposals

getProposals call



See Proposals Addresses

well in so far there will only be 1 address

call to Ballot.getProposals

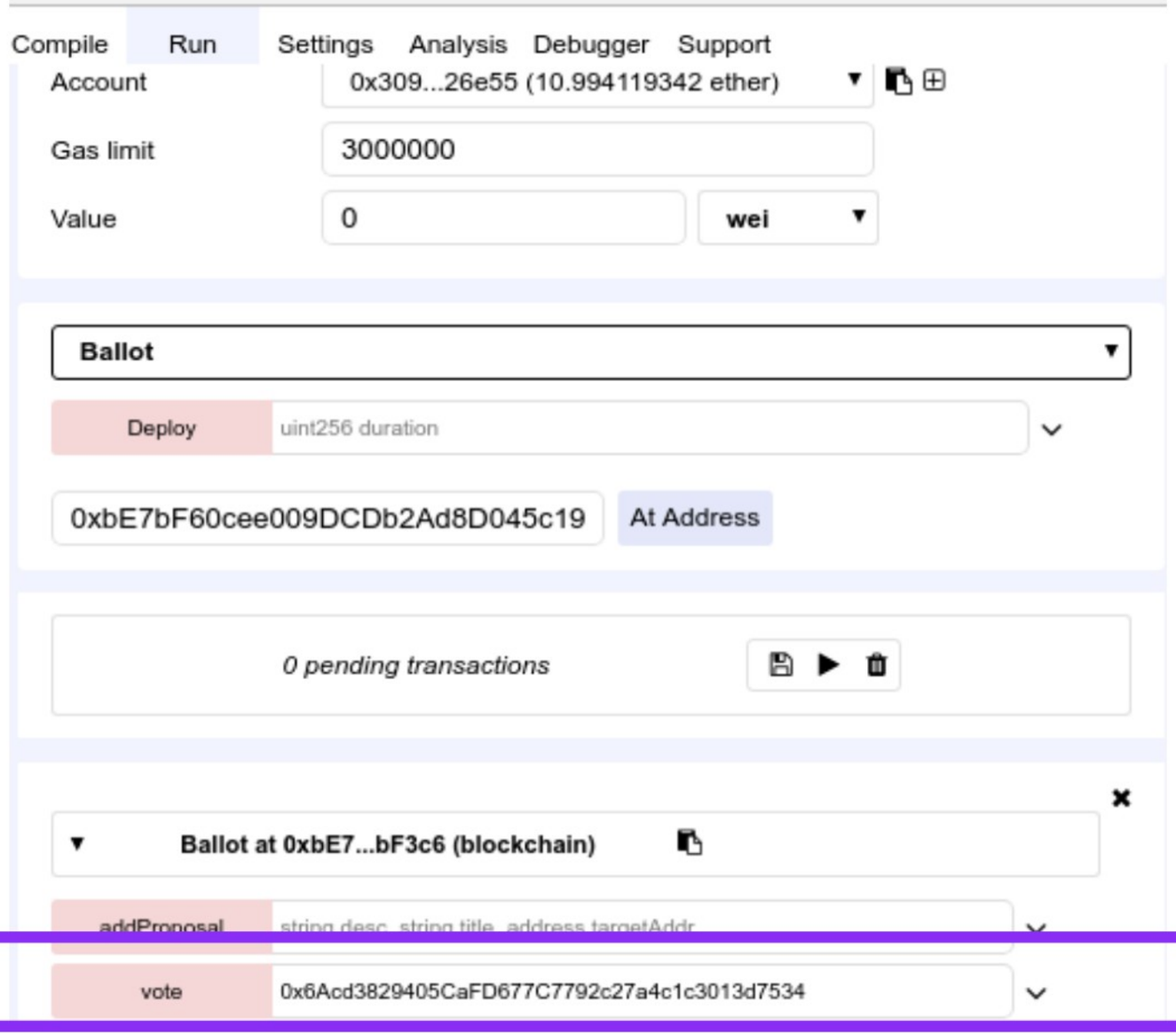
▾[call] **from:**0x3092232fb25e6b359a9fead9ed07ad752ff26e55 **to:**Ballot.getProposals()
data:0x625...64c48

Debug

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

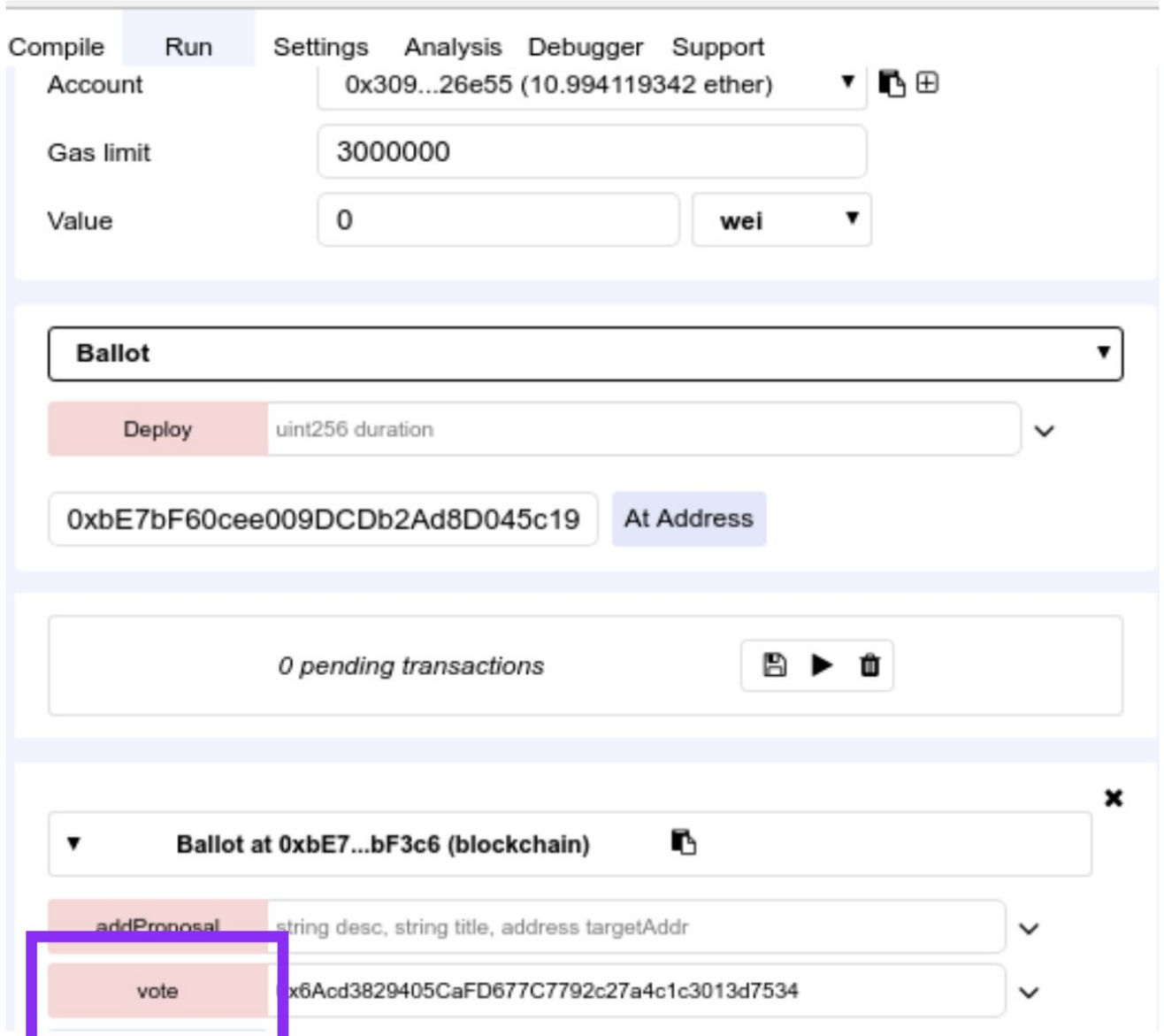
Vote for one Proposal

Paste Proposal Address you want to vote for



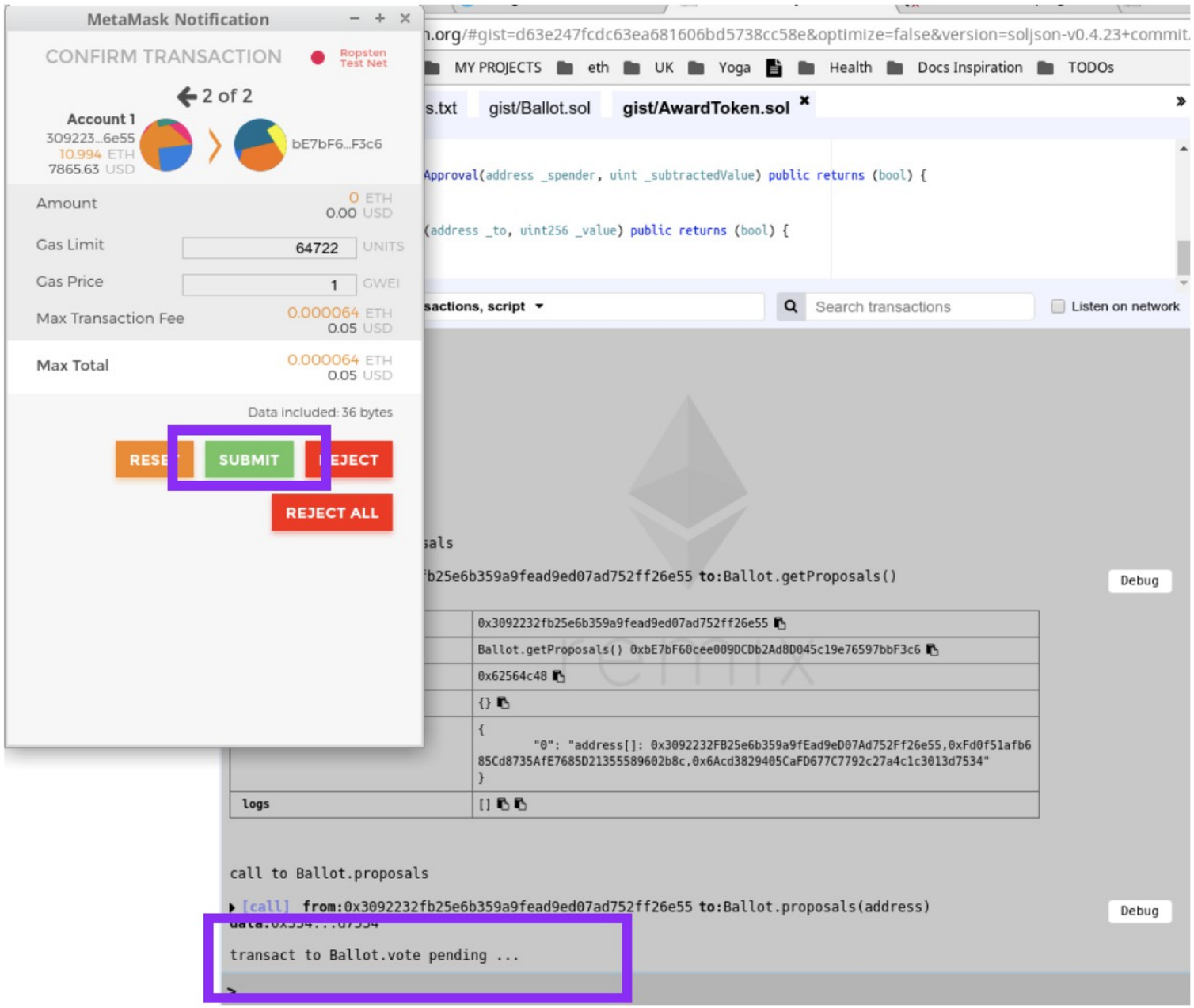
Execute vote transaction

vote button



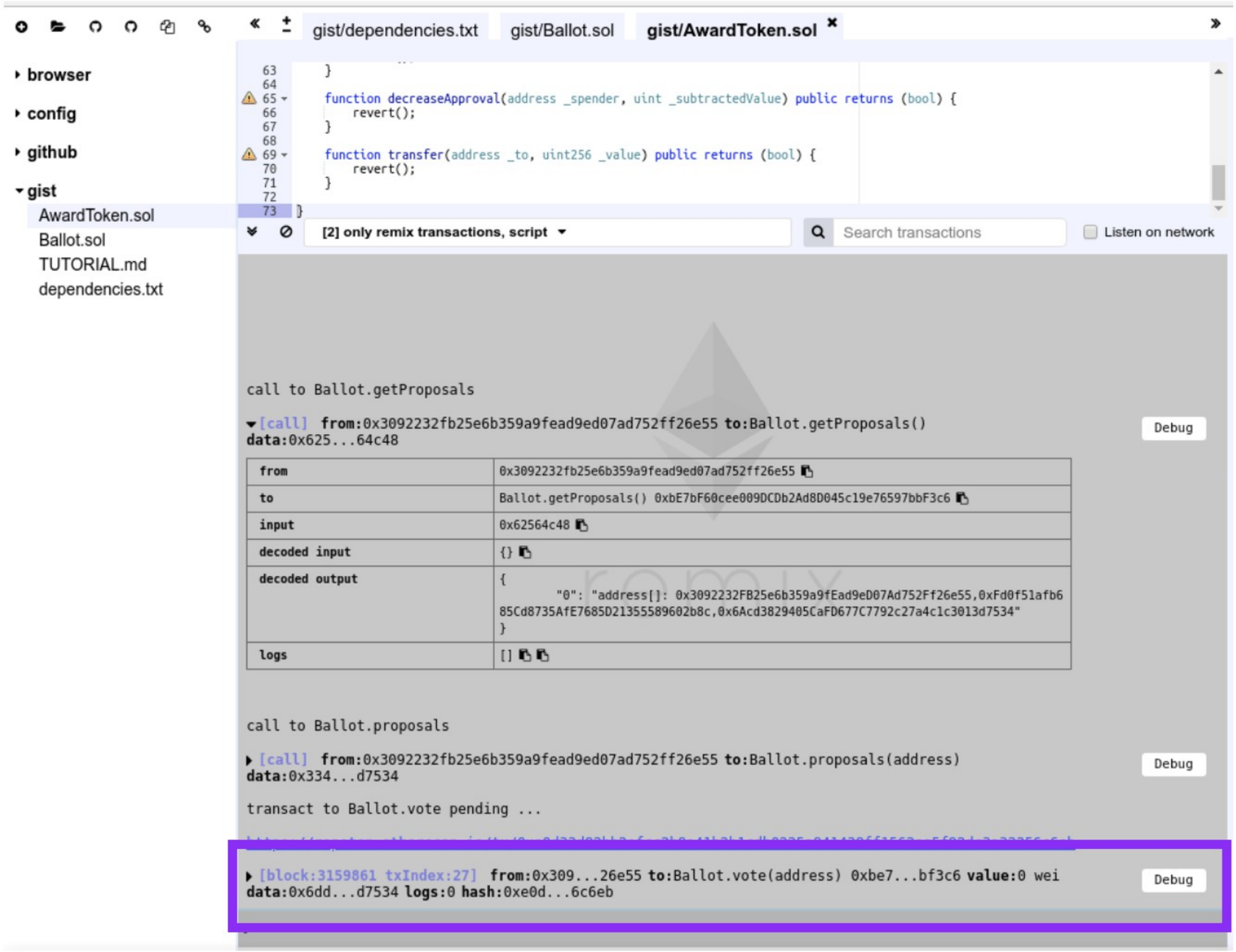
Confirm the transaction

Submit button



Check if tx succeeded

Terminal logs in Remix



Now let's try it out connecting a frontend

http://bit.ly/remix-voting

(when dependencies.js is the active file)

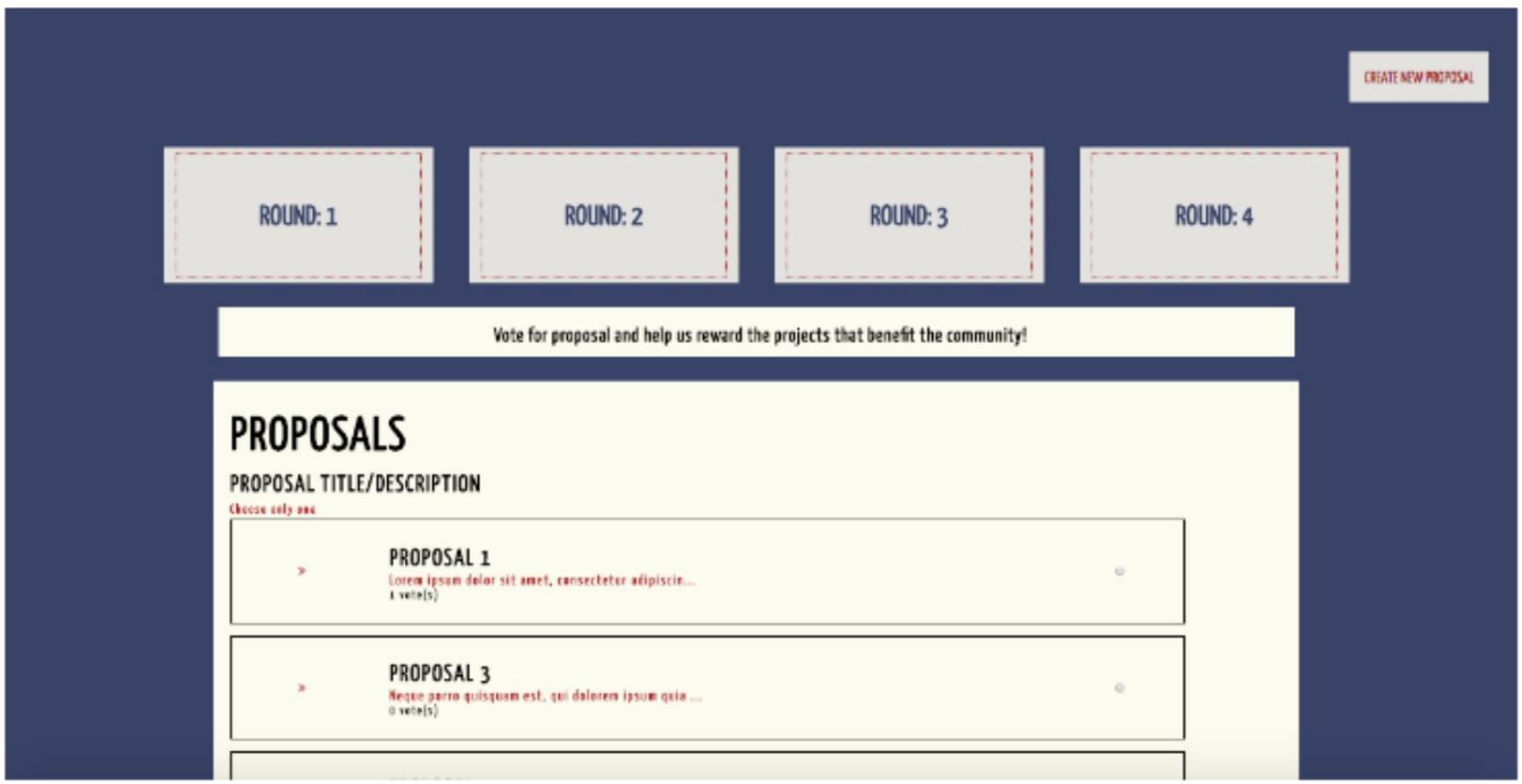
To access our Award Token from this frontend - you need the address of the Award Token.

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;
```

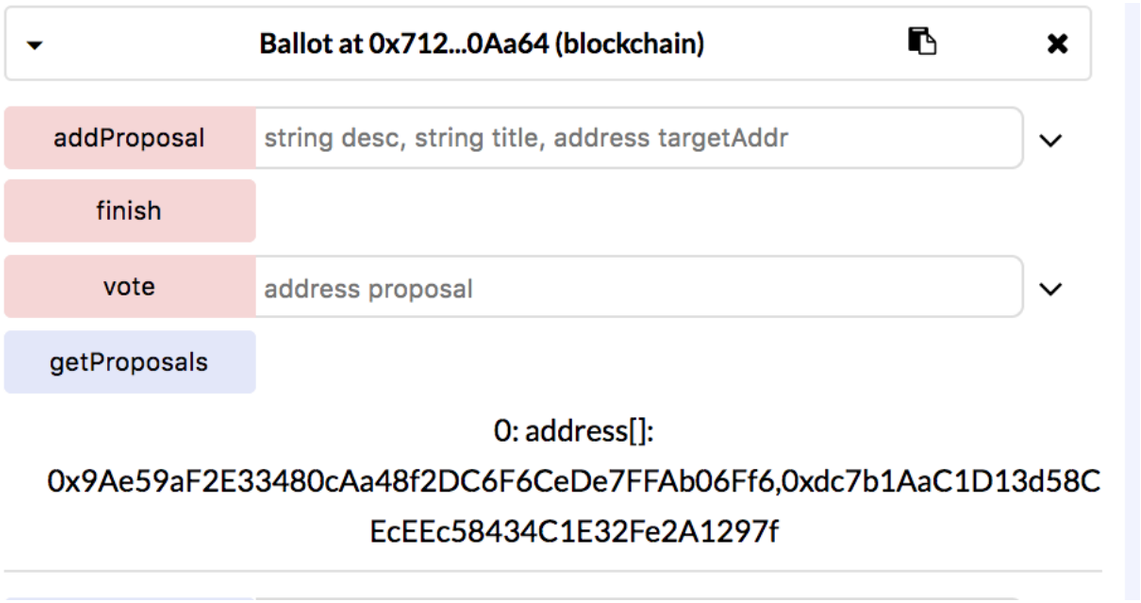
```
contract AwardToken is MintableToken {
    uint quantity;
    uint ballotPeriod = 7 hours;
    Ballot public currBallot;
    address[] public prevWinners;
```



Let's check results

http://bit.ly/remix-voting

Check the state of the contract



2 proposals have been added

Thank you

@ninabreznik @ryestew @yann300
@serapath @lianahusikyan @iurimatias

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