Multiple ways of loading files intro to Remix

NINA BREZNIK

@ninabreznik

YANN LEVREAU

@ninabreznik

IURI MATIAS

@iurimatias

ROB STUPAY

@ryestew

ALEX PRAETORIUS

@SERAPATH

LIANA HUSIKYAN

@LianaHus

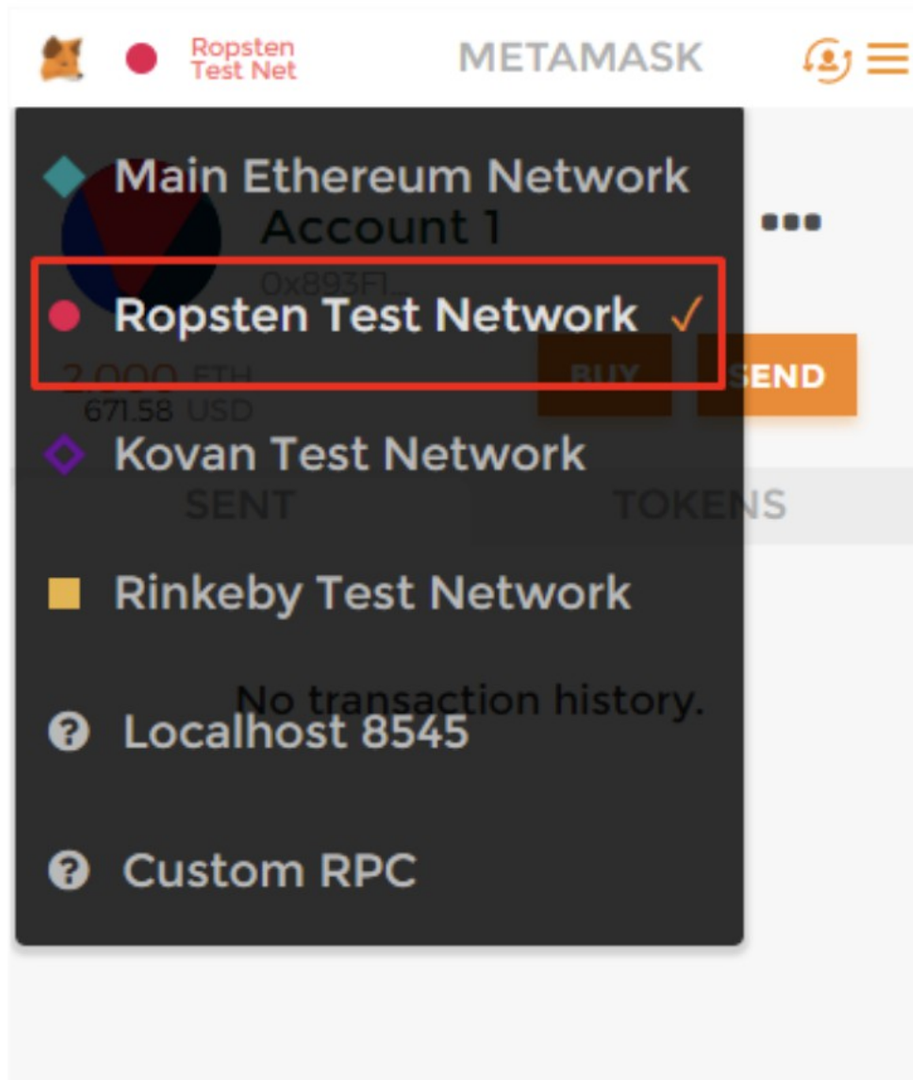
Part 1

Tour, Loading Contracts, & Compiling

This tutorial assumes somes knowledge of Remix.

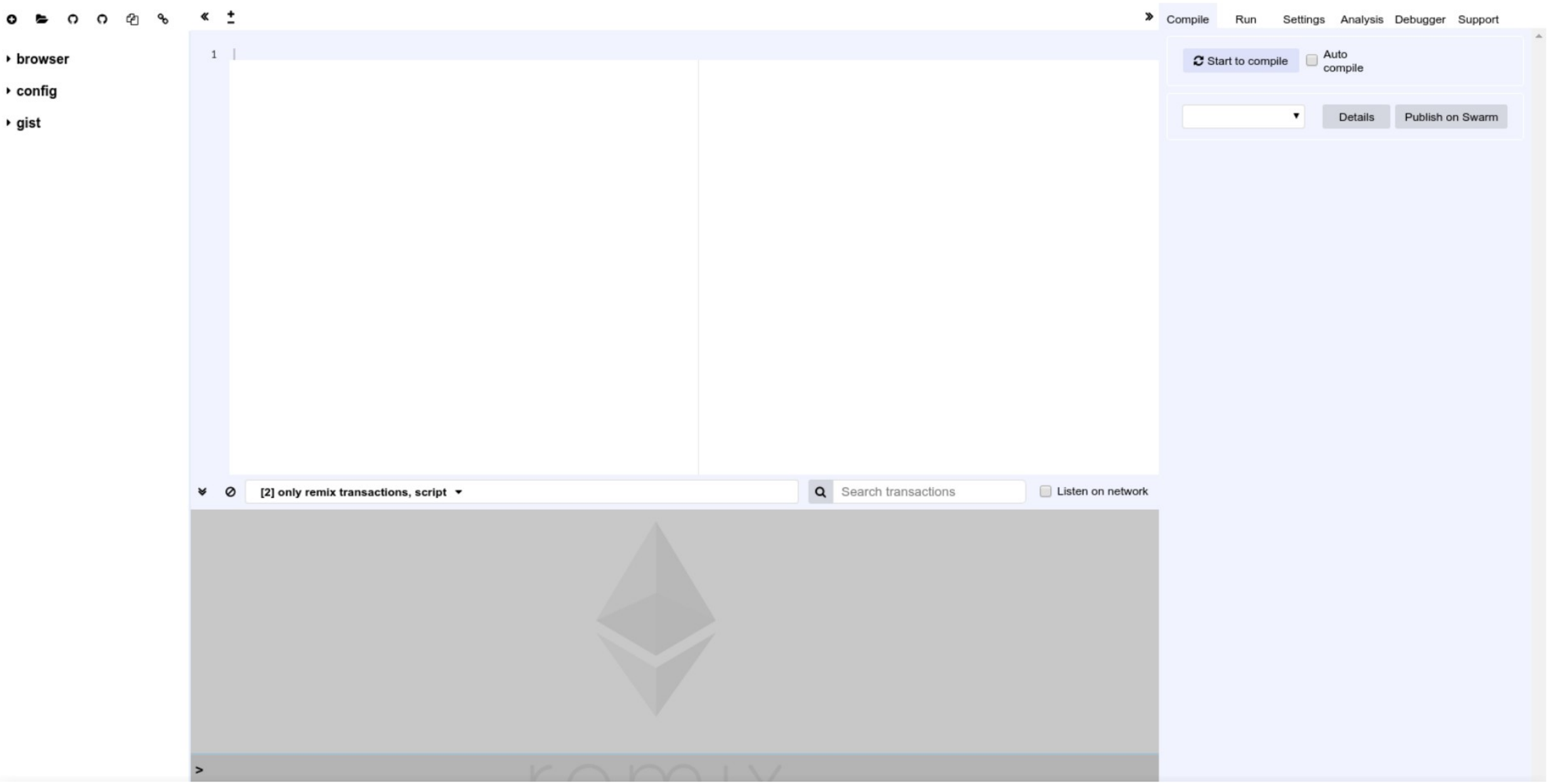
Login to Metamask

Robsten Test Network



Let’s get started

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



Remix Tour

https://remix-alpha.ethereum.org

File Explorer

- browser
  - AwardToken.sol
  - Ballot2.sol
  - Ballot_orig.sol
  - Donation.sol
  - README.md
  - multiSig2.sol
  - multisig.sol
  - multisig1.sol
  - scenario.json
  - setup.txt
- config

Editor

browser/Ballot_orig.sol

```
pragma solidity ^0.4.0;
contract Ballot {

    struct Voter {
        uint weight;
        bool voted;
        uint8 vote;
        address delegate;
    }
    struct Proposal {
        uint voteCount;
    }

    address chairperson;
    mapping(address => Voter) voters;
    Proposal[] proposals;
```

Compile Tab (active)

Compile | Run | Settings | Analysis | Debugger | Support

Start to compile | Auto compile | Hide warnings

Ballot

Details | Publish on Swarm | ABI | Bytecode

Static Analysis raised 2 warning(s) that requires your attention.
Click here to show the warning(s).

```
browser/Ballot_orig.sol:19:5: Warning: Defining c
    function Ballot(uint8 _numProposals) public
    ^ (Relevant source part starts here and spans
```

[2] only remix transactions, script | Search transactions

Terminal

```
 - Welcome to Remix v0.6.4 -

You can use this terminal for:
 - Checking transactions details and start debugging.
 - Running JavaScript scripts.
 - Running JavaScript scripts involving web3 if the current environement is injected p
rovider or Web3 provider.
 - Executing common command to interact with the Remix interface (see list of commands
 below). Note that these command can also be included in a JavaScript script.

remix.debug(hash): Start debugging a transaction.

remix.loadgist(id): Load a gist in the file explorer.

remix.loadurl(url): Load the given url in the file explorer. The url can be of type gi
thub, swarm or ipfs.

remix.setproviderurl(url): Change the current provider to Web3 provider and set the ur
l endpoint.

remix.exeCurrent(): Run the script currently displayed in the editor

remix.help(): Display this help message
```

Console

>

Remix Commands

https://remix-alpha.ethereum.org

browser
- AwardToken.sol
- Ballot2.sol
- Ballot_orig.sol
- Donation.sol
- README.md
- multiSig2.sol
- multisig.sol
- multisig1.sol
- scenario.json
- setup.txt

config

browser/Ballot_orig.sol

```
pragma solidity ^0.4.0;
contract Ballot {

    struct Voter {
        uint weight;
        bool voted;
        uint8 vote;
        address delegate;
    }
    struct Proposal {
        uint voteCount;
    }

    address chairperson;
    mapping(address => Voter) voters;
    Proposal[] proposals;
```

Compile | Run | Settings | Analysis | Debugger | Support | Tes

Start to compile | Auto compile | Hide warnings

Ballot

Details | Publish on Swarm | ABI | Bytecode

Static Analysis raised 2 warning(s) that requires your attention.
Click here to show the warning(s).

```
browser/Ballot_orig.sol:19:5: Warning: Defining c
    function Ballot(uint8 _numProposals) public {
    ^ (Relevant source part starts here and spans
```

[2] only remix transactions, script | Search transactions

```
 - Welcome to Remix v0.6.4 -

You can use this terminal for:
 - Checking transactions details and start debugging.
 - Running JavaScript scripts.
 - Running JavaScript scripts involving web3 if the current environement is injected p
rovider or Web3 provider.
 - Executing common command to interact with the Remix interface (see list of commands
 below). Note that these command can also be included in a JavaScript script.

remix.debug(hash): Start debugging a transaction.

remix.loadgist(id): Load a gist in the file explorer.

remix.loadurl(url): Load the given url in the file explorer. The url can be of type gi
thub, swarm or ipfs.

remix.setproviderurl(url): Change the current provider to Web3 provider and set the ur
l endpoint.

remix.exeCurrent(): Run the script currently displayed in the editor

remix.help(): Display this help message

>
```

Set environment

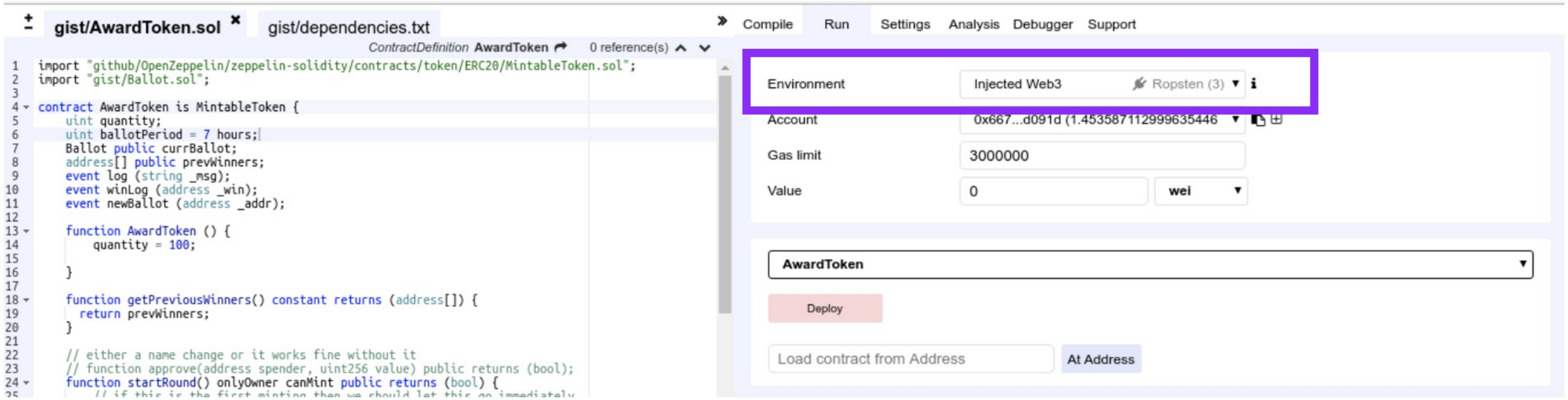
Run tab-> Environment:

Javascript VM - a simple and quick environment – only visible to your machine

Injected web3 (Ropsten) – or whatever testnet you have setup in metamask

Web3 Provider

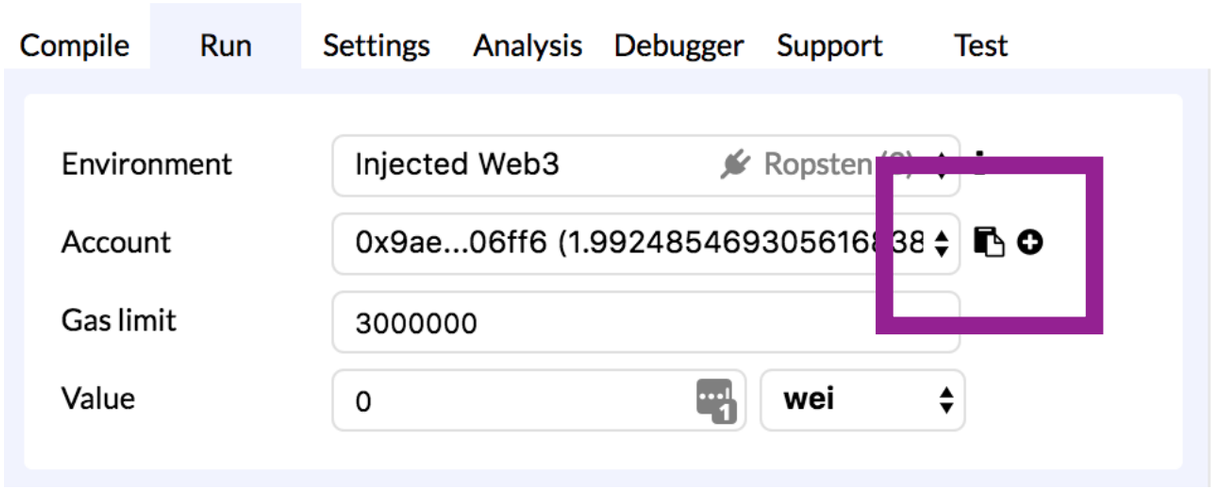
Choose: **Injected web3 (Ropsten)**



Get some TEST ether

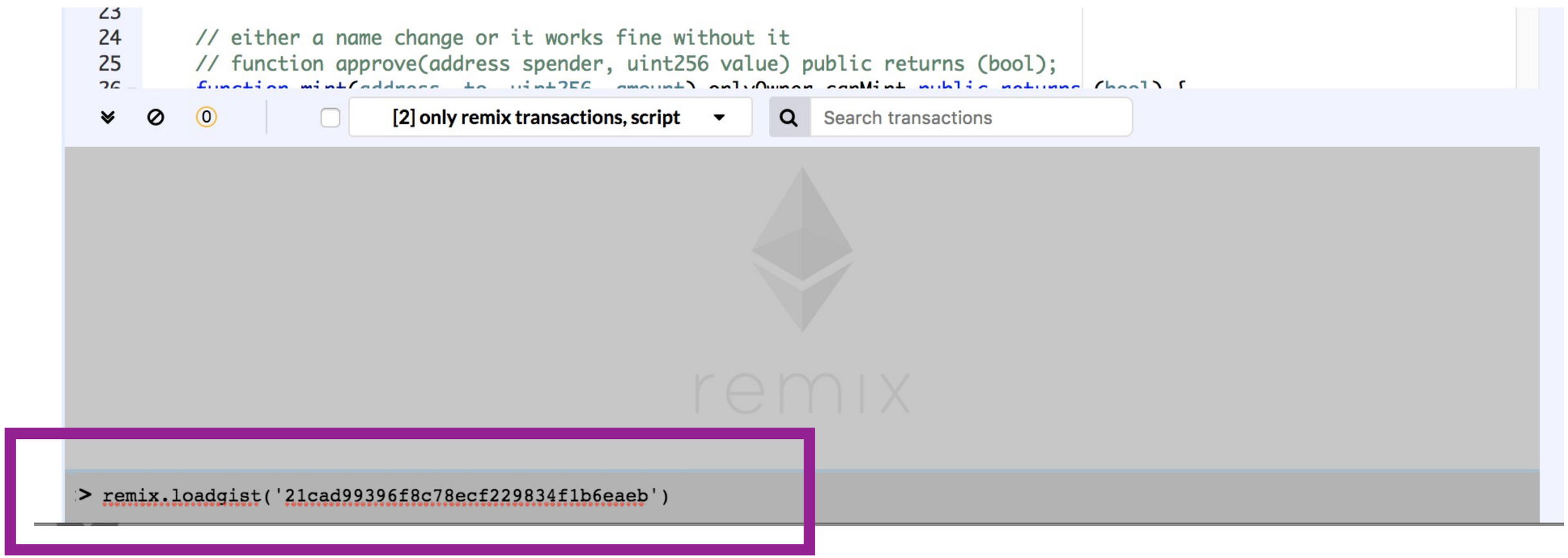
http://faucet.ropsten.be:3001/

But FIRST:
Copy your address - here or in Metamask



Load files from a GIST

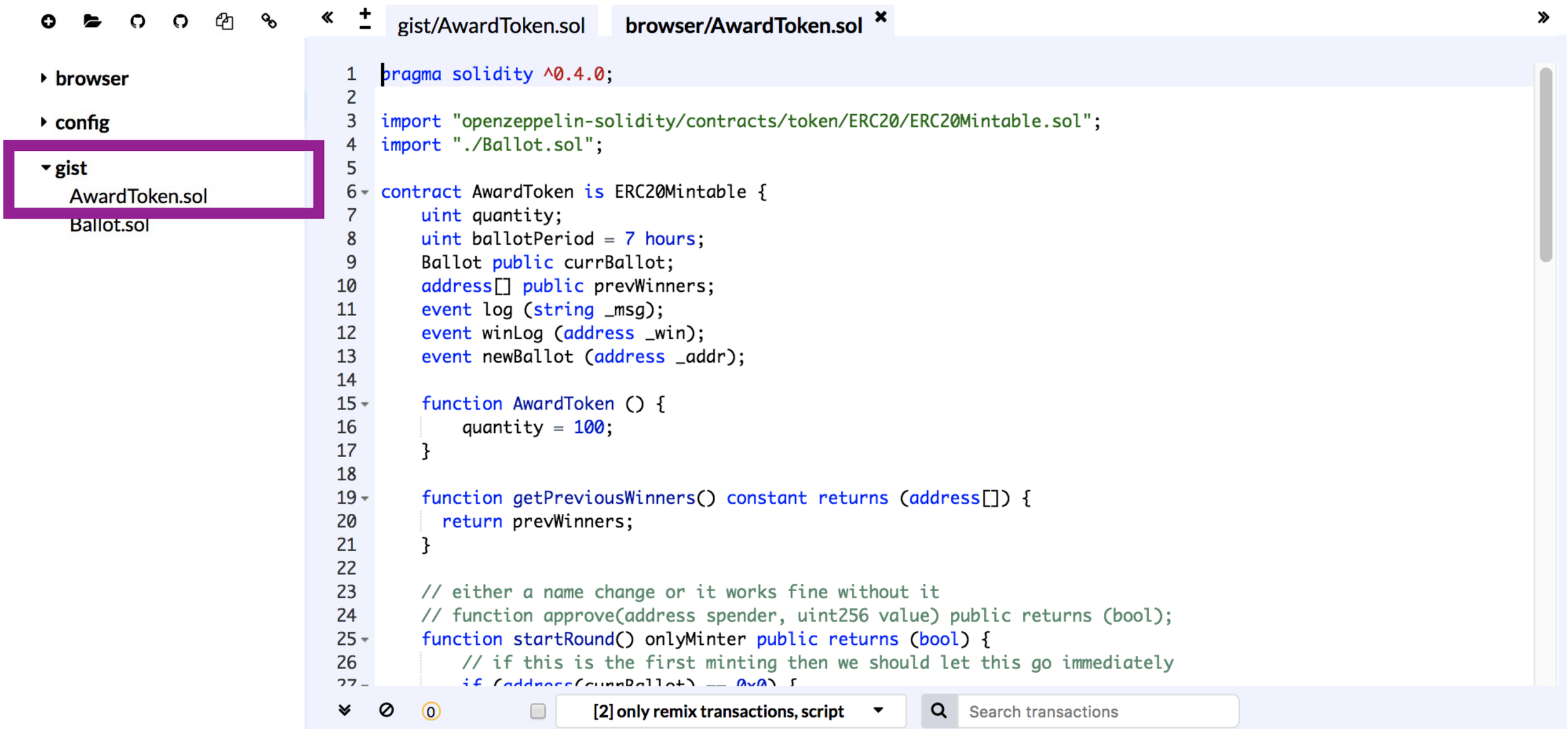
remix.loadgist('21cad99396f8c78ecf229834f1b6eaeb')



here in the console

Open file

gist/AwardToken



Setup for remixd:

To access local files on your computer and npm modules

In a Terminal / Console
(not inside remix but on your computer)

cd to a directory where you want to install the npm modules.
(typically your project's directory)

Type these commands:

```
npm init -y
npm install remixd
npm install -E openzeppelin-solidity
```

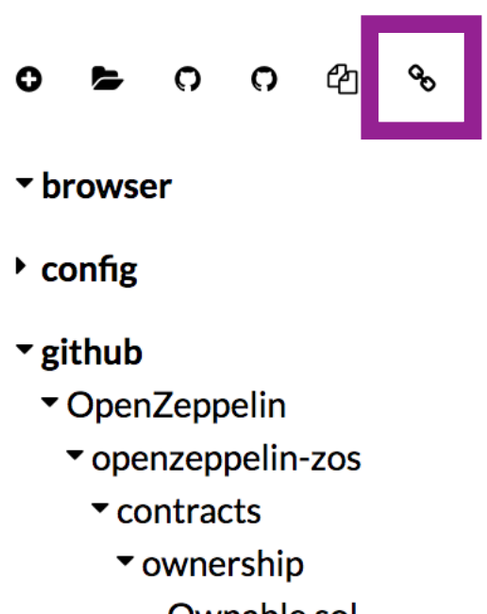
Get Remixd Going!

In a Terminal / Console
And in your project’s directory

Type:

remixd -s ./

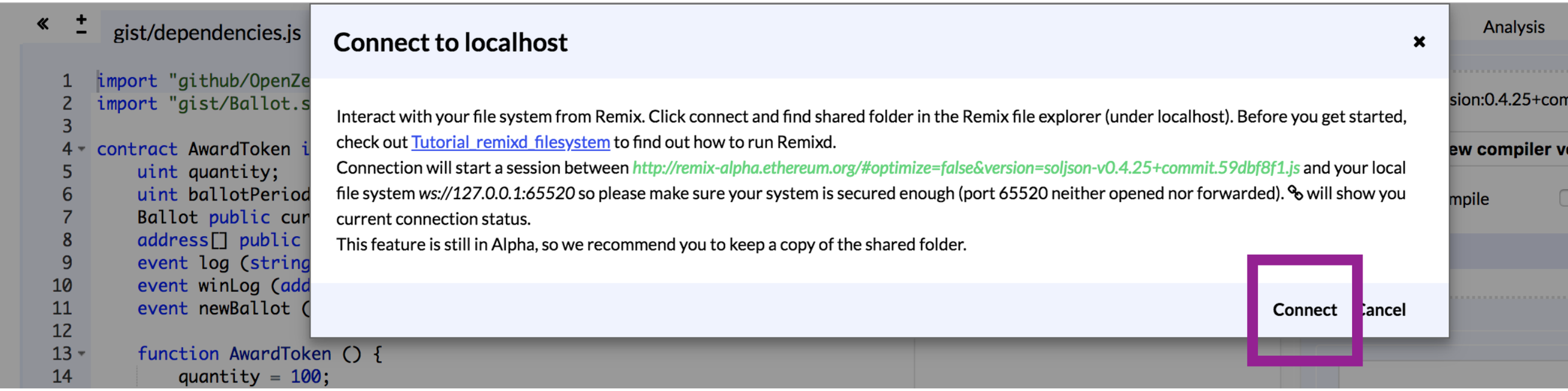
Then go to remix and turn on remixd by clicking the link icon at the top of the File Explorer (top left of Remix)



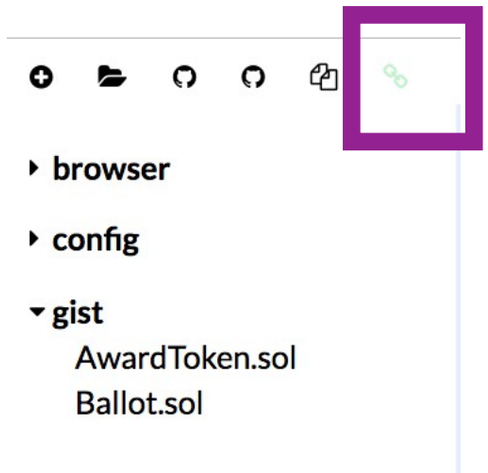
Open file

gist/AwardToken

Click ‘Connect’



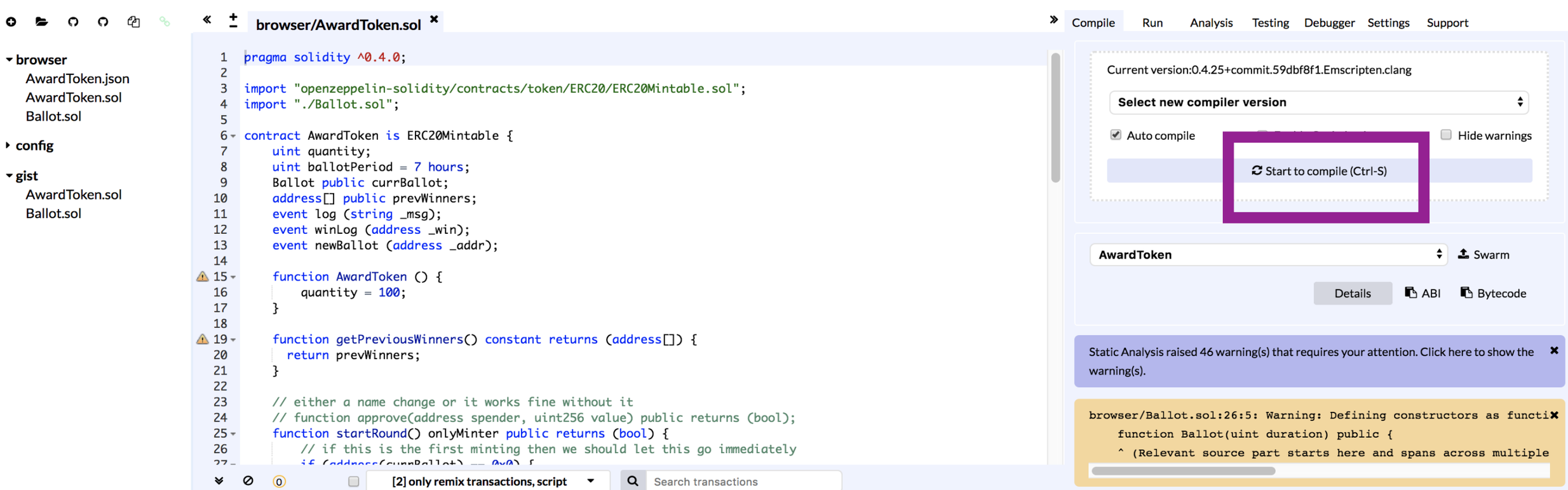
The link icon should turn light green.



Compile AwardToken

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

Click the “Start to compile” button



If you have no compile errors (in pink) ...

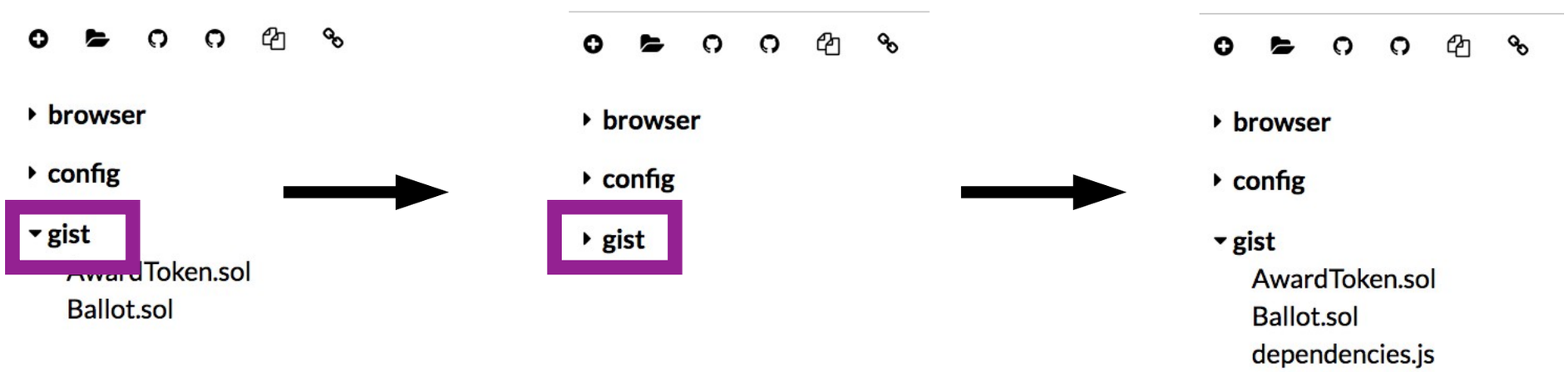
SUCCESS!!

Load from GITHUB

In the console type:

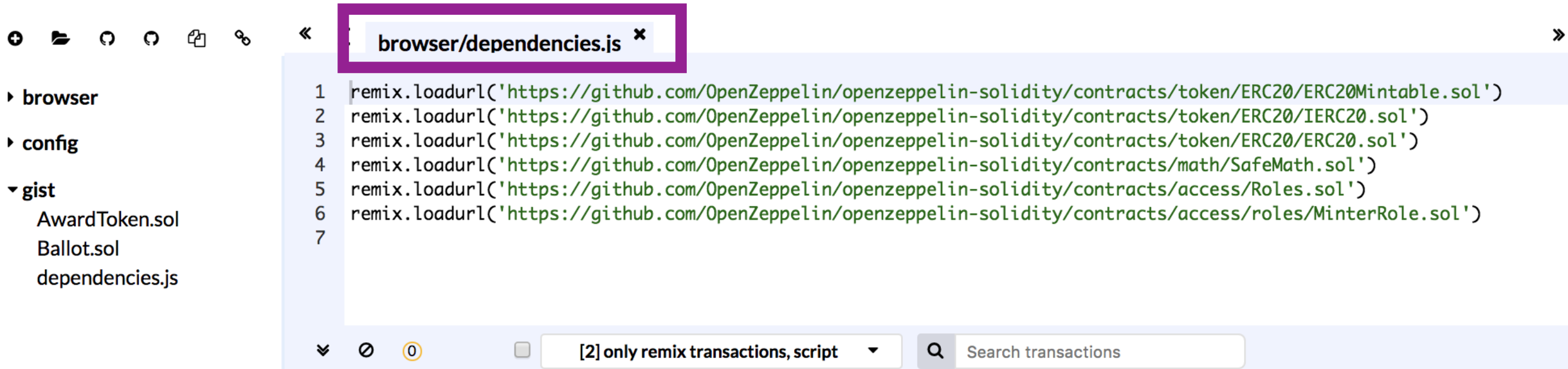
remix.loadgist('1d3250ca532b5f1e3ed4f52b18feb602')

To see the new file loaded you need to open and close the caret next to gist :



Load up the dependencies

1. Click dependencies.js to make it the active file
(it will appear in the editor)



2. In the console type: remix.exeCurrent()
-or-
copy each line of the dependencies.js into the console to load each file of the github repo

3. Open and close the caret next to githubto see the loaded files.

Update the code to point to the github repos

1. Go to AwardToken and update line 3

Replace:
import "openzeppelin-solidity/contracts/token/ERC20/ERC20Mintable.sol";

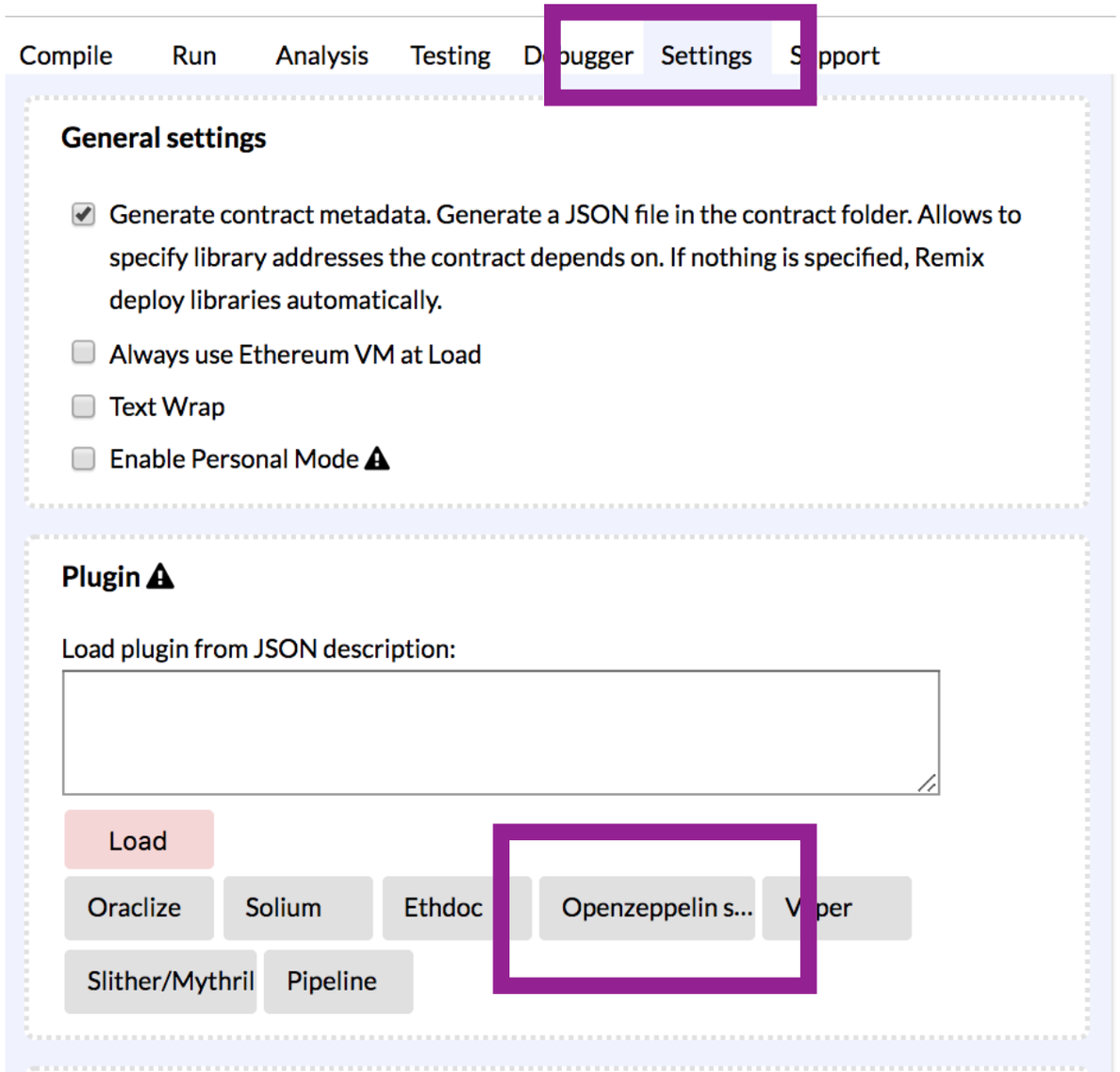
With:
import "github/OpenZeppelin/openzeppelin-solidity/contracts/token/ERC20/ERC20Mintable.sol";

Now you are pointing to the downloaded repo in remix.

Hit the “start to compile” button on the compile tab.
And there shouldn't be any errors… (famous last words).

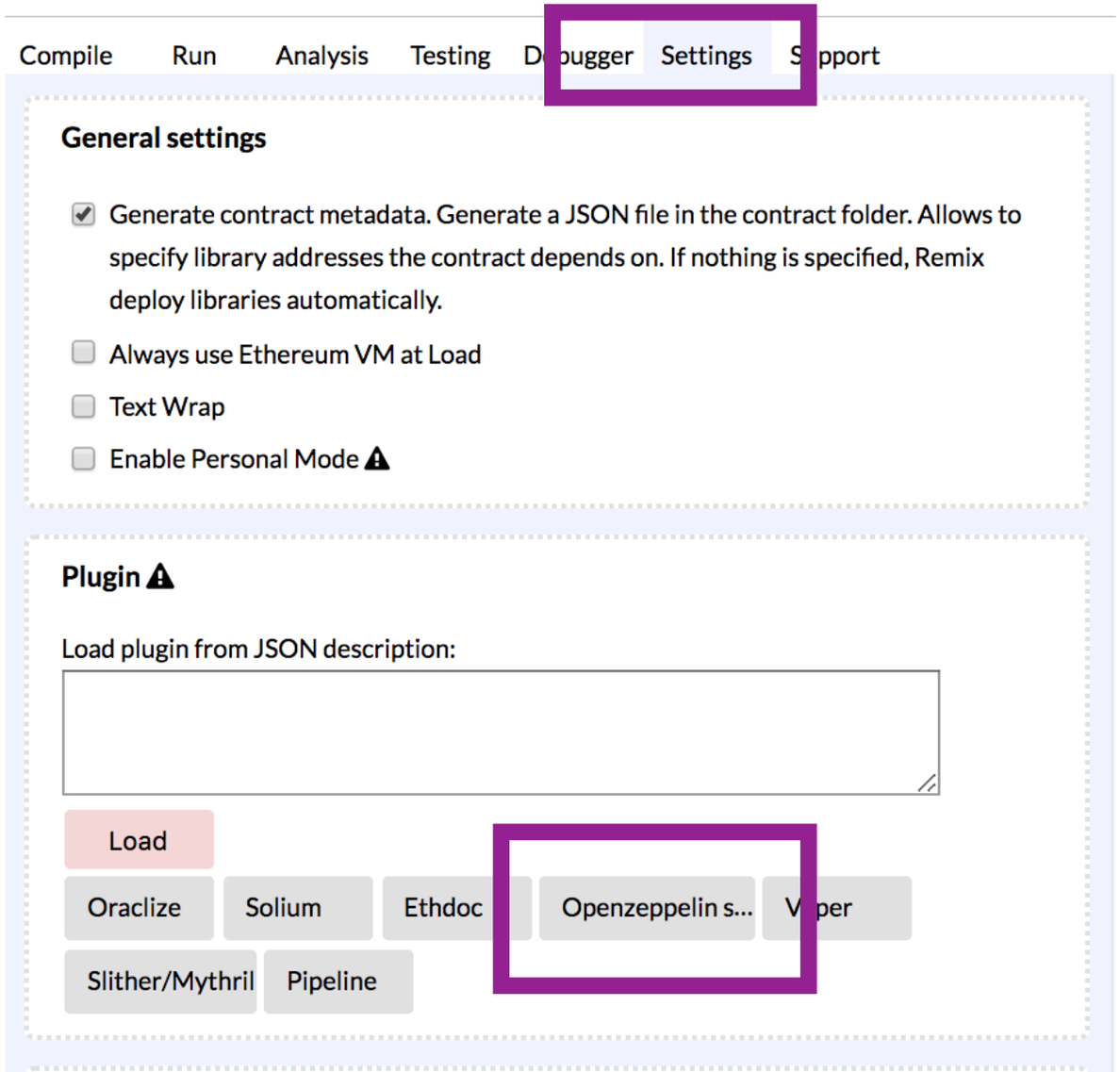
Load from a url – via the Zeppelin plugin

Click the Openzeppelin plugin button in the settings tab



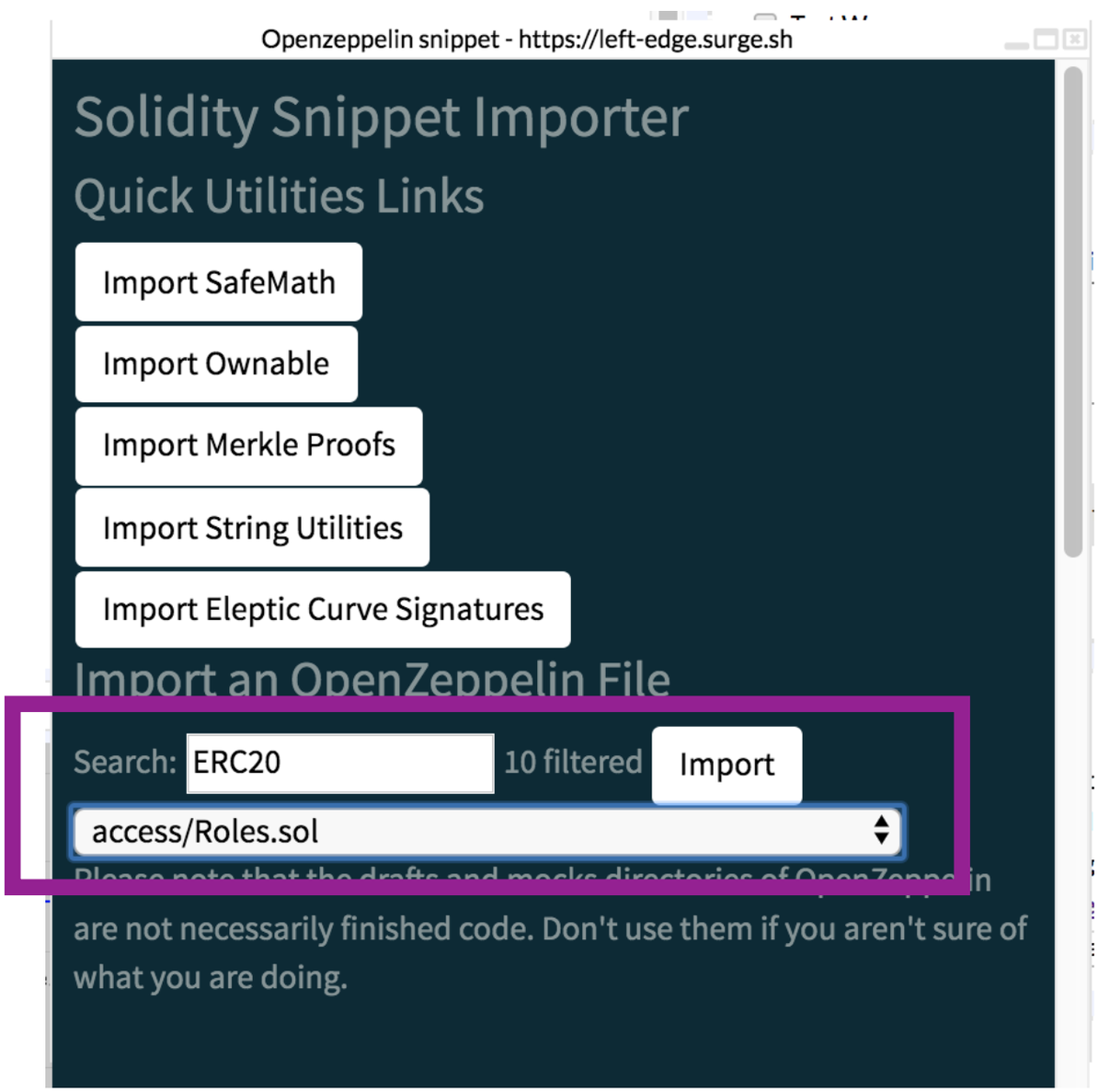
Load from a url – via the Zeppelin plugin

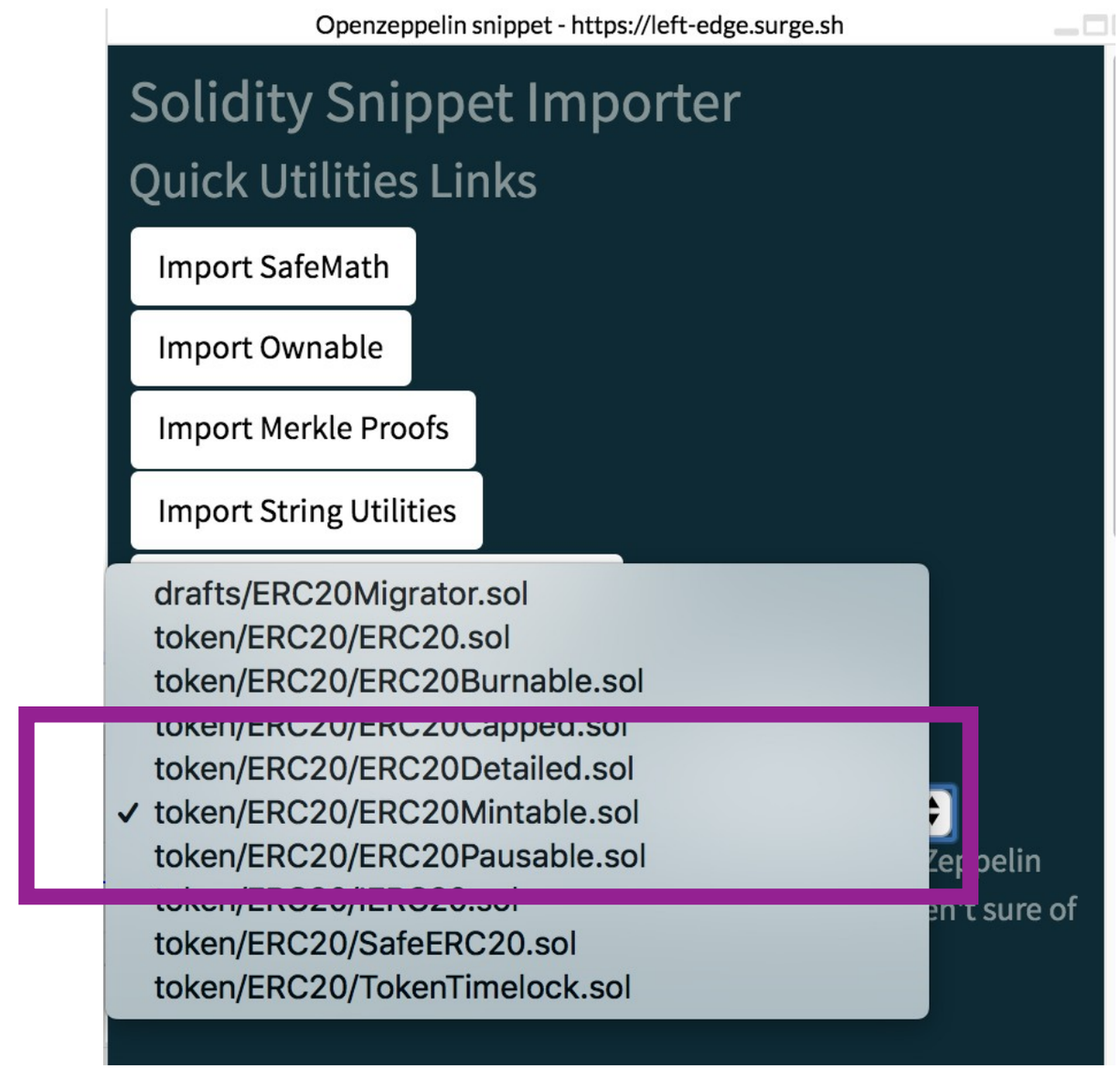
Click the Openzeppelin plugin button in the settings tab



Choose ERC20Mintable.sol

1. In the window that pops up – input: **ERC20** in the Search box.
2. In the select box – choose ERC20Mintable.sol & click Import





Update AwardToken.sol

The Zeppelin plugin has written a new line in AwardToken that imports ERC20Mintable.sol via https.

So, delete this line:

import "github/OpenZeppelin/openzeppelin-solidity/contracts/token/ERC20/ERC20Mintable.sol";

AwardToken's import lines should look like this:

gist/AwardToken.sol

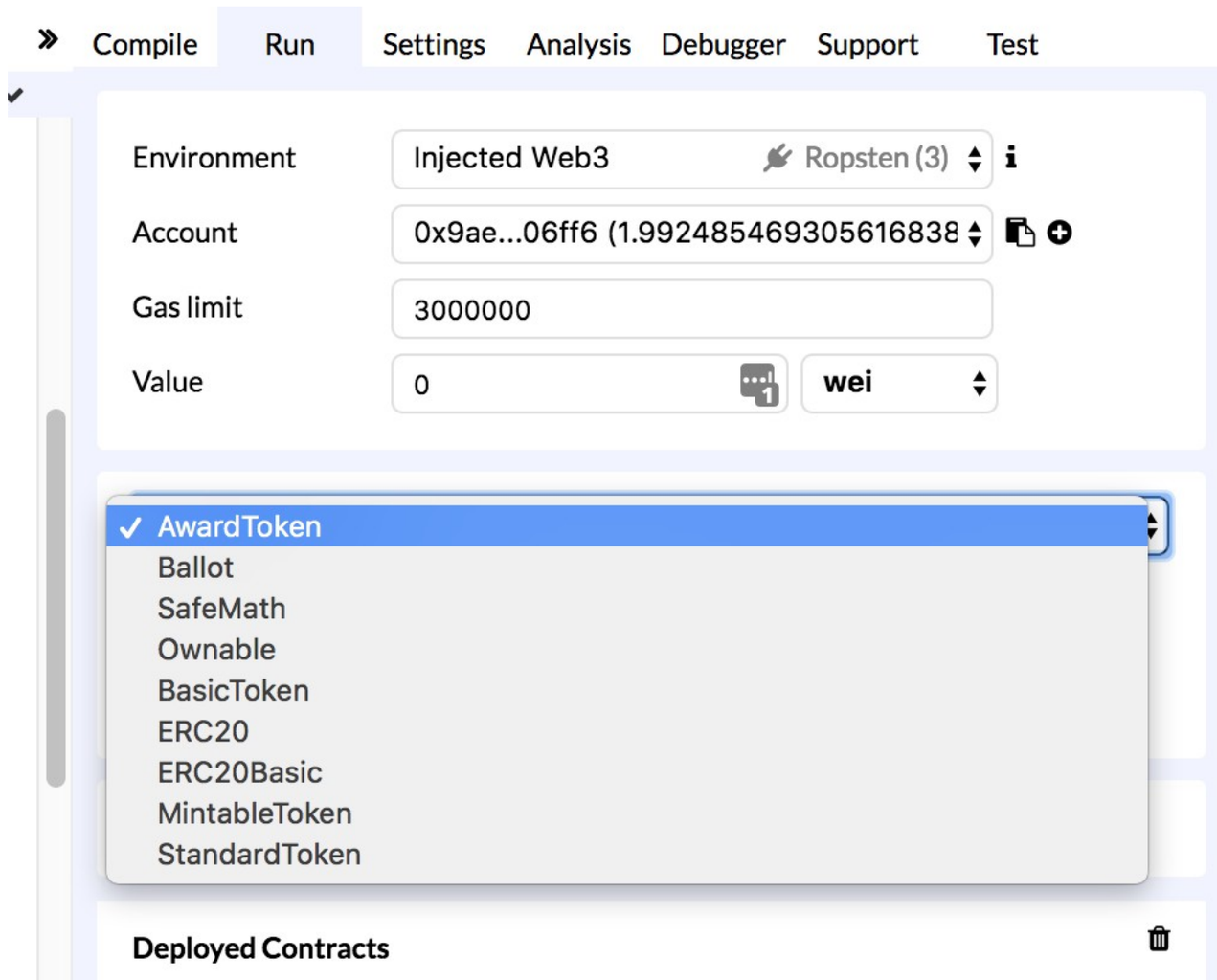
```
pragma solidity ^0.4.0;
import "https://github.com/OpenZeppelin/openzeppelin-solidity/contracts/token/ERC20/ERC20Mintable.sol";

import "./Ballot.sol";

```

Then hit the compile button and it should work... (again famous last words)

See compiled contracts

AwardToken + all dependencies

Imported Contracts

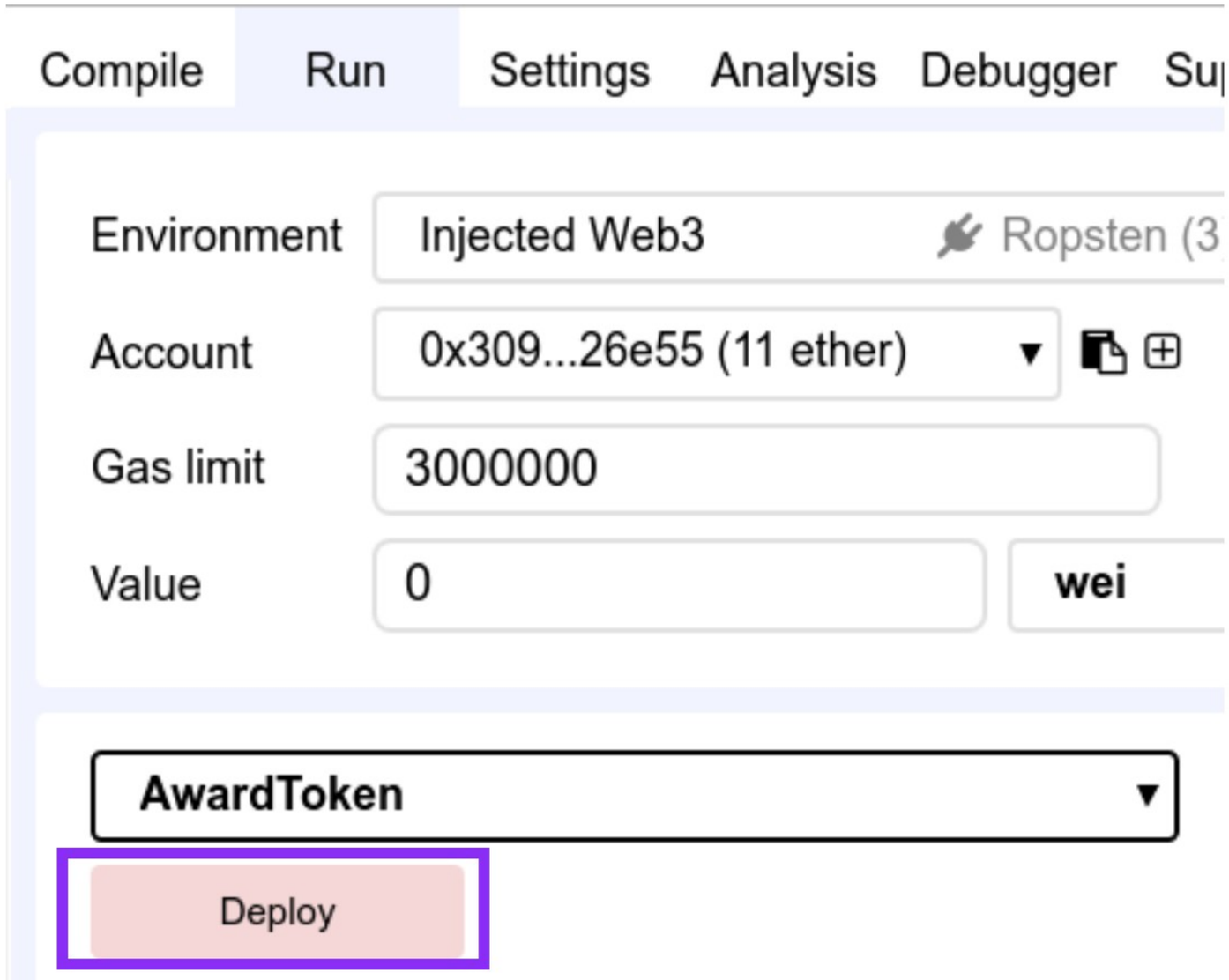


Part II

Deploy & AtAddress

Deploy the contract

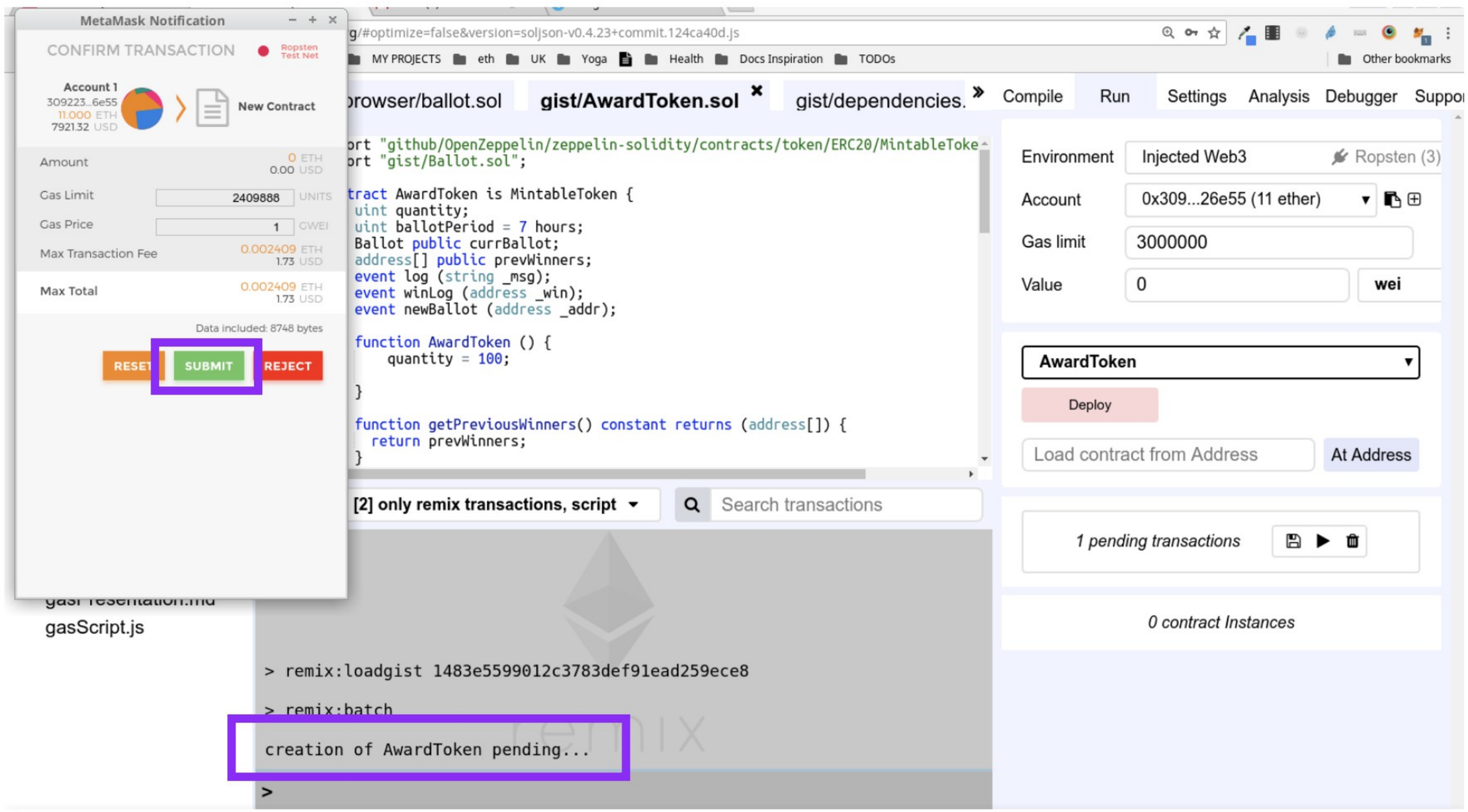
Run tab: Deploy button



Confirm the transaction

Submit button

But make sure you put in a gas price!



Check if tx is mined

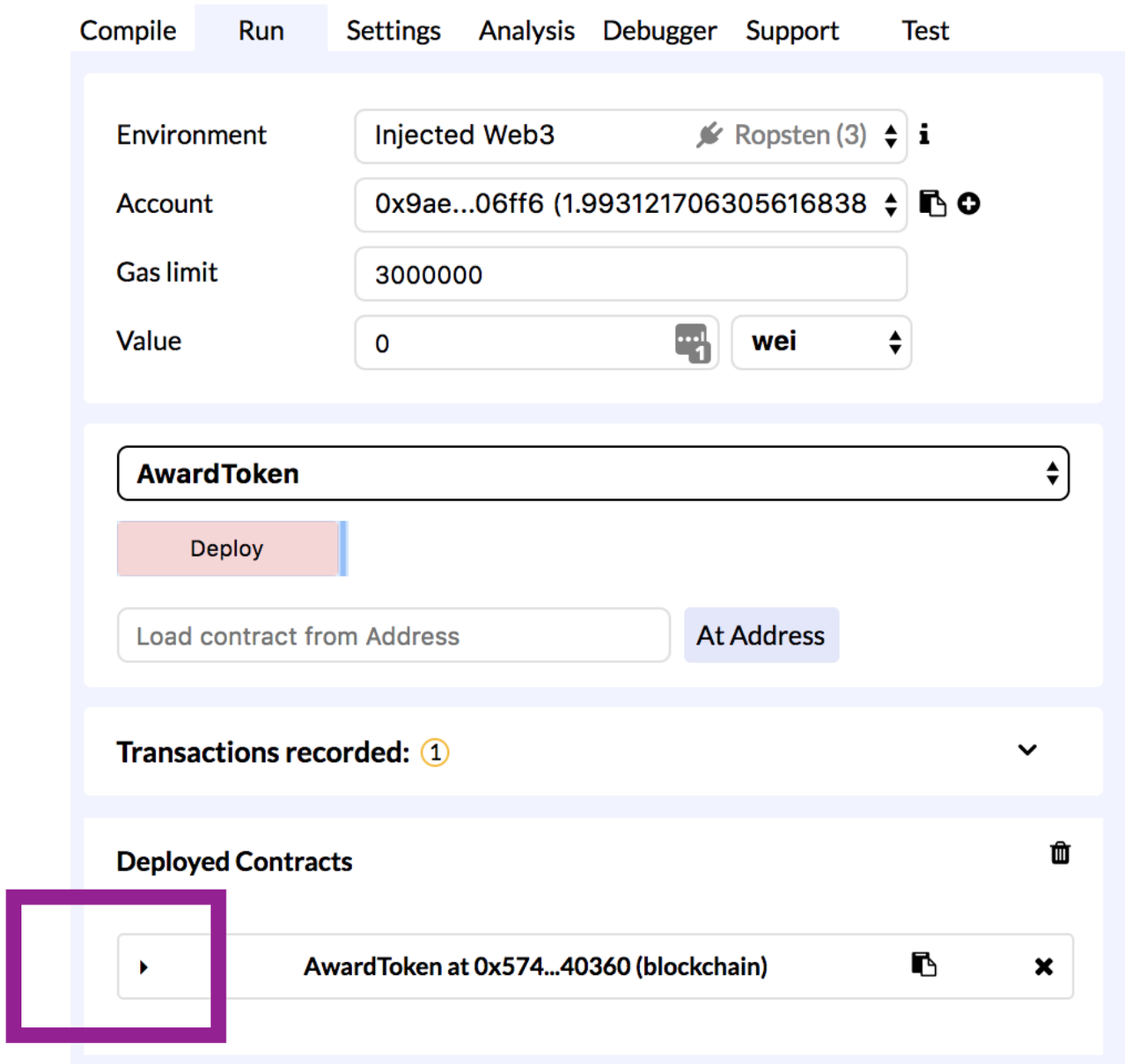
Terminal logs in Remix

```
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



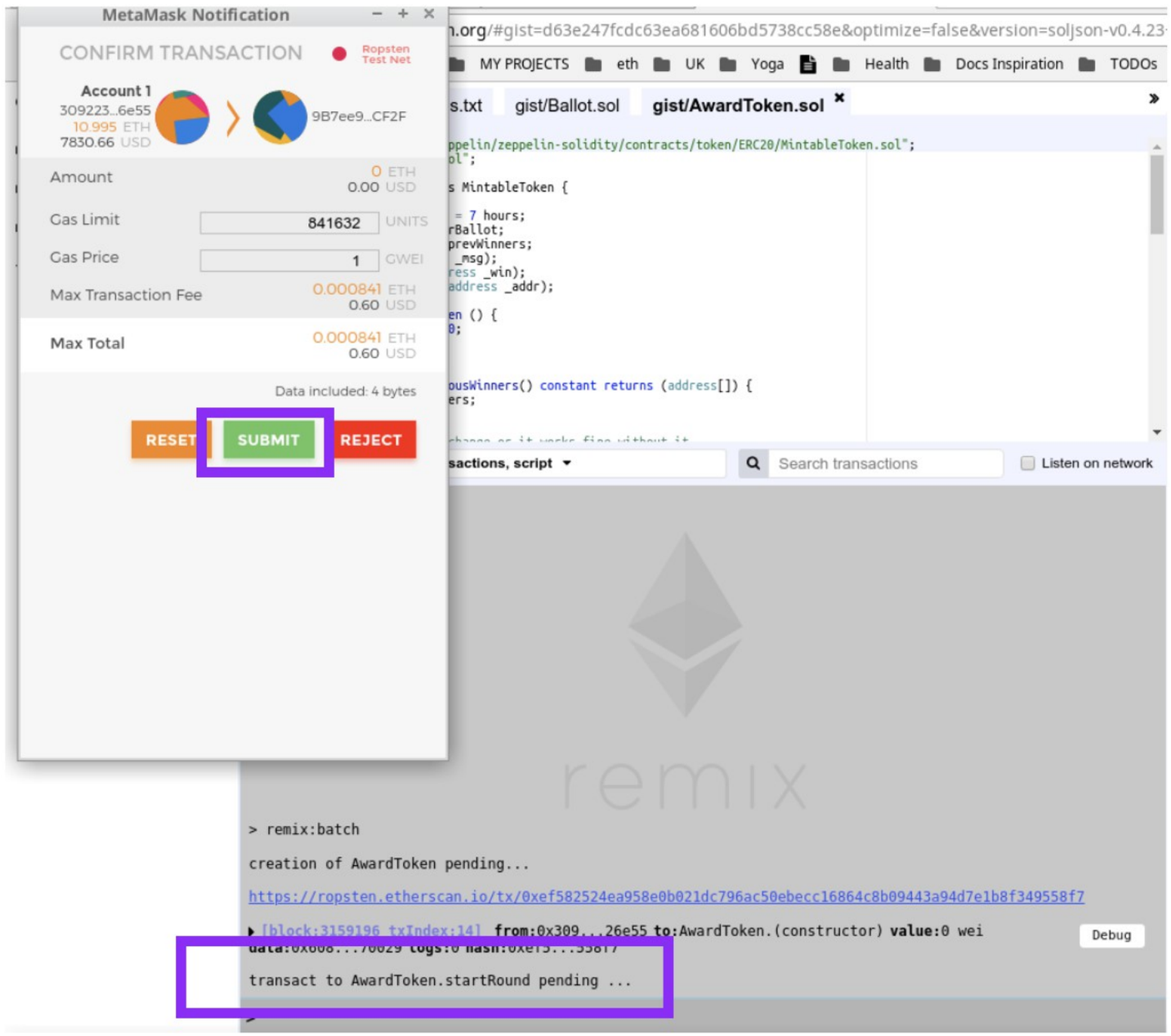
Behold!

The Interactive UI for AwardToken.sol contract

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

```
transact to AwardToken.startRound pending ...
https://ropsten.etherscan.io/tx/0x5a97b4946979f52dfb6dc8ab2fecebb8fd43515ff4e25597ecb9d0a88472c8b2
▸ [block:3159300 txIndex:12]  from:0x309...26e55 to:AwardToken.startRound() 0x9b7...0cf2f
value:0 wei data:0x55e...3f086 logs:1 hash:0x5a9...2c8b2
```

Debug

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

Take a look at the startRound function in the editor

– just to see what it is doing...

gist/dependencies.js gist/AwardToken.sol

```
    // function approve(address spender, uint256 value) public returns (bool);
    function startRound() onlyOwner canMint public returns (bool) {
        // if this is the first minting then we should let this go immediately
        if (address(currBallot) == 0x0) {
            currBallot = new Ballot(ballotPeriod);
            newBallot(currBallot);
        } else {
            revert();
        }
    }

    function closeRoundEarly () onlyOwner {
        if (address(currBallot) != 0x0 && !currBallot.timeOut()) {
          currBallot.finish();
        } else revert();
    }

    function closeRound() onlyOwner {
        //  this can only be done by the owner of the contract

        if (address(currBallot) != 0x0 && currBallot.timeOut()) {
            // get winner
            address winner = currBallot.winningProposal();
            winLog(winner);
            // send to winner - but first make sure the address is valid
            if ( winner == 0x0){
```

Get ballot's address

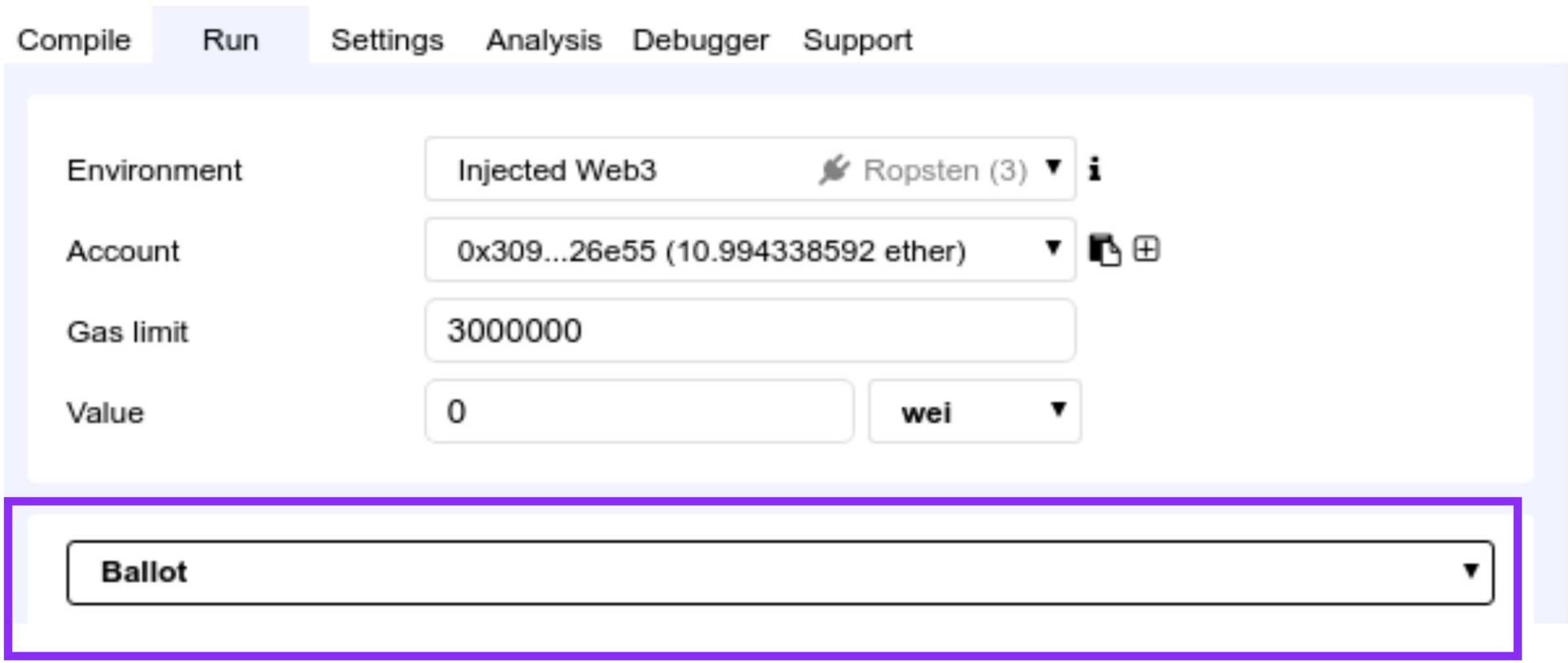
Execute currBallot call

Copy ballot's address

currBallot output

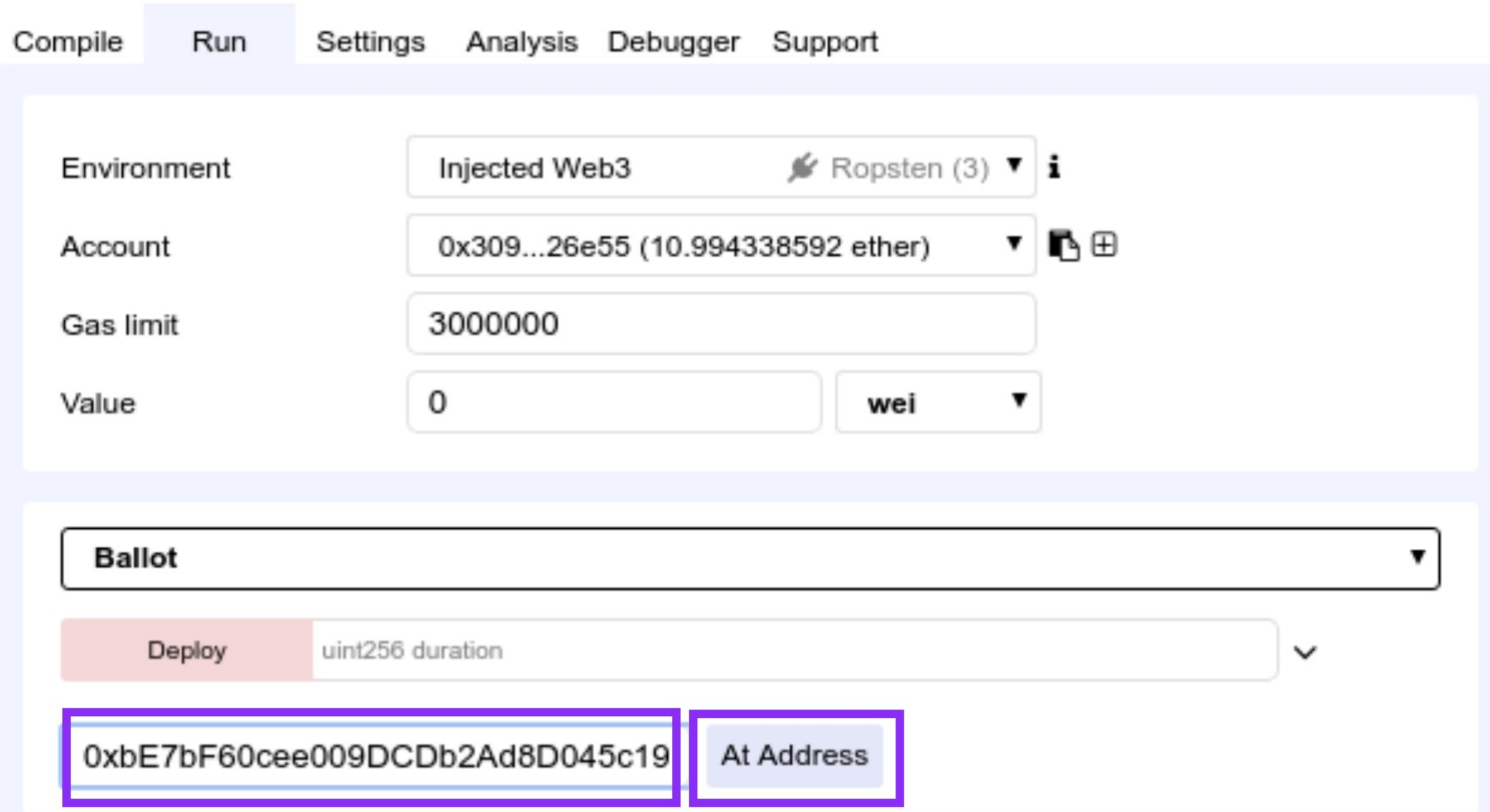
Switch to Ballot

Run tab: dropdown



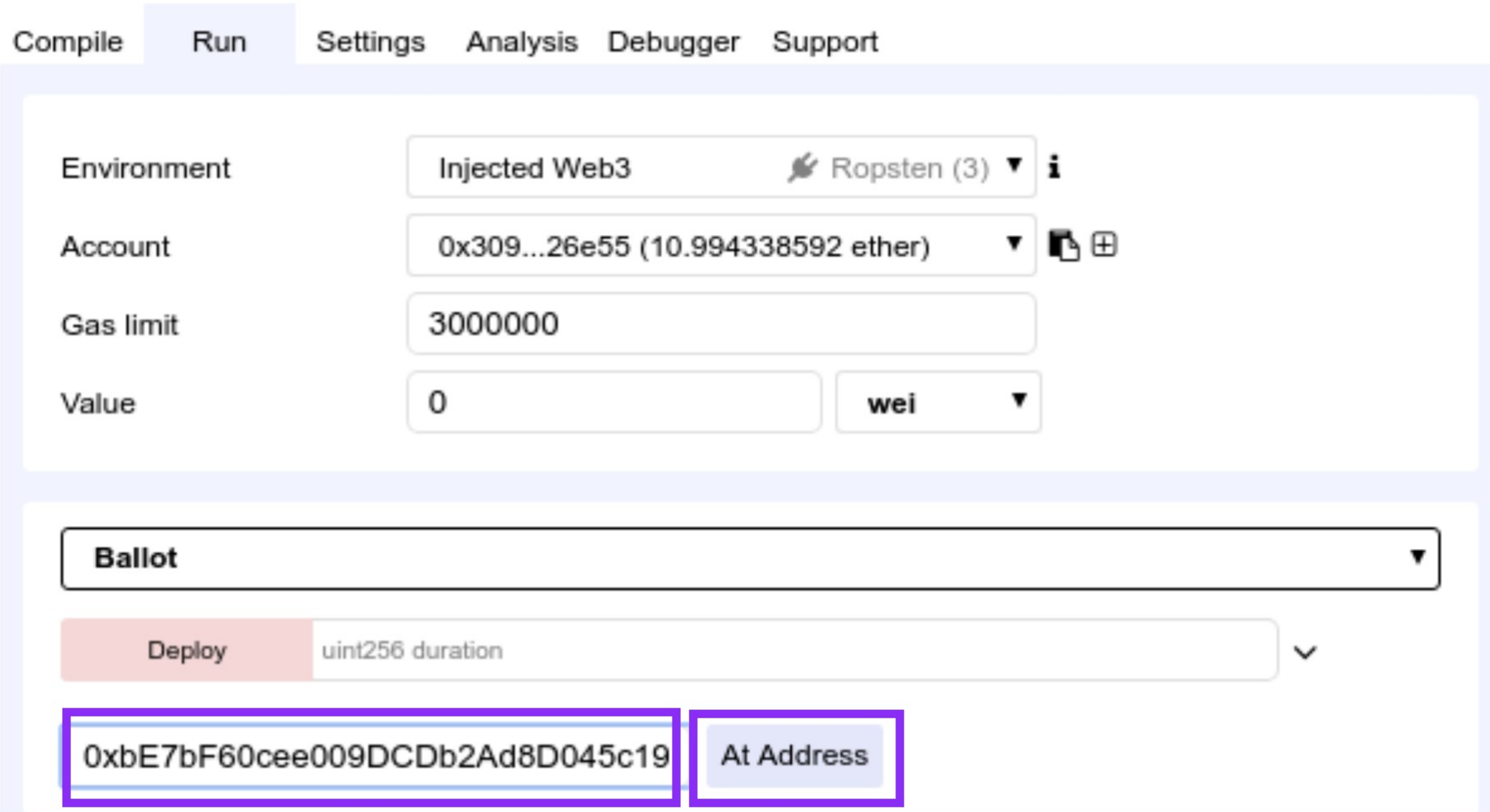
Access Ballot contract

Paste address + click At Address



Access Ballot contract

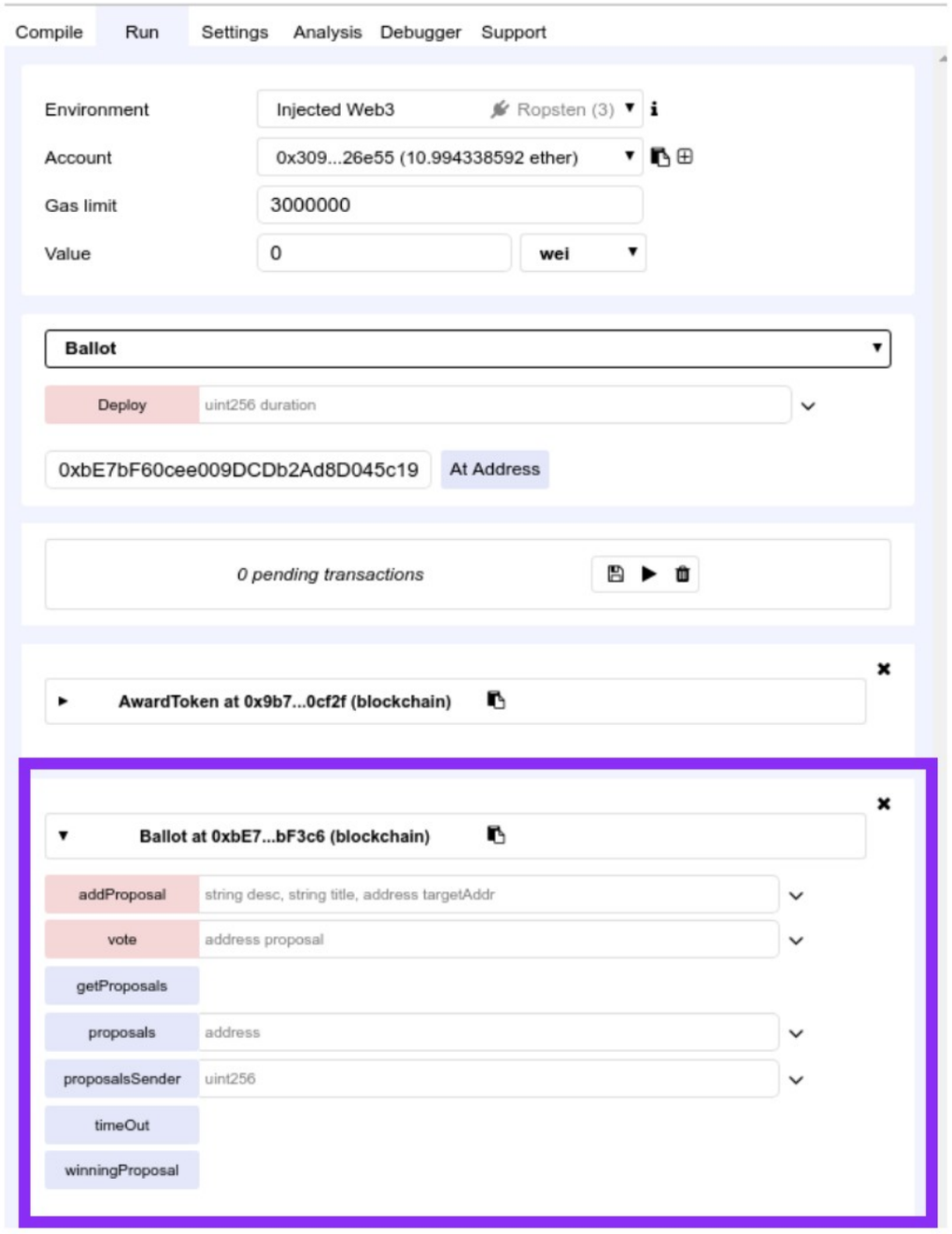
Paste address + click At Address



See autogenerated UI

(you might need to scroll down)

- Interact with all the functions in the contract
- And all the functions in the inherited contracts

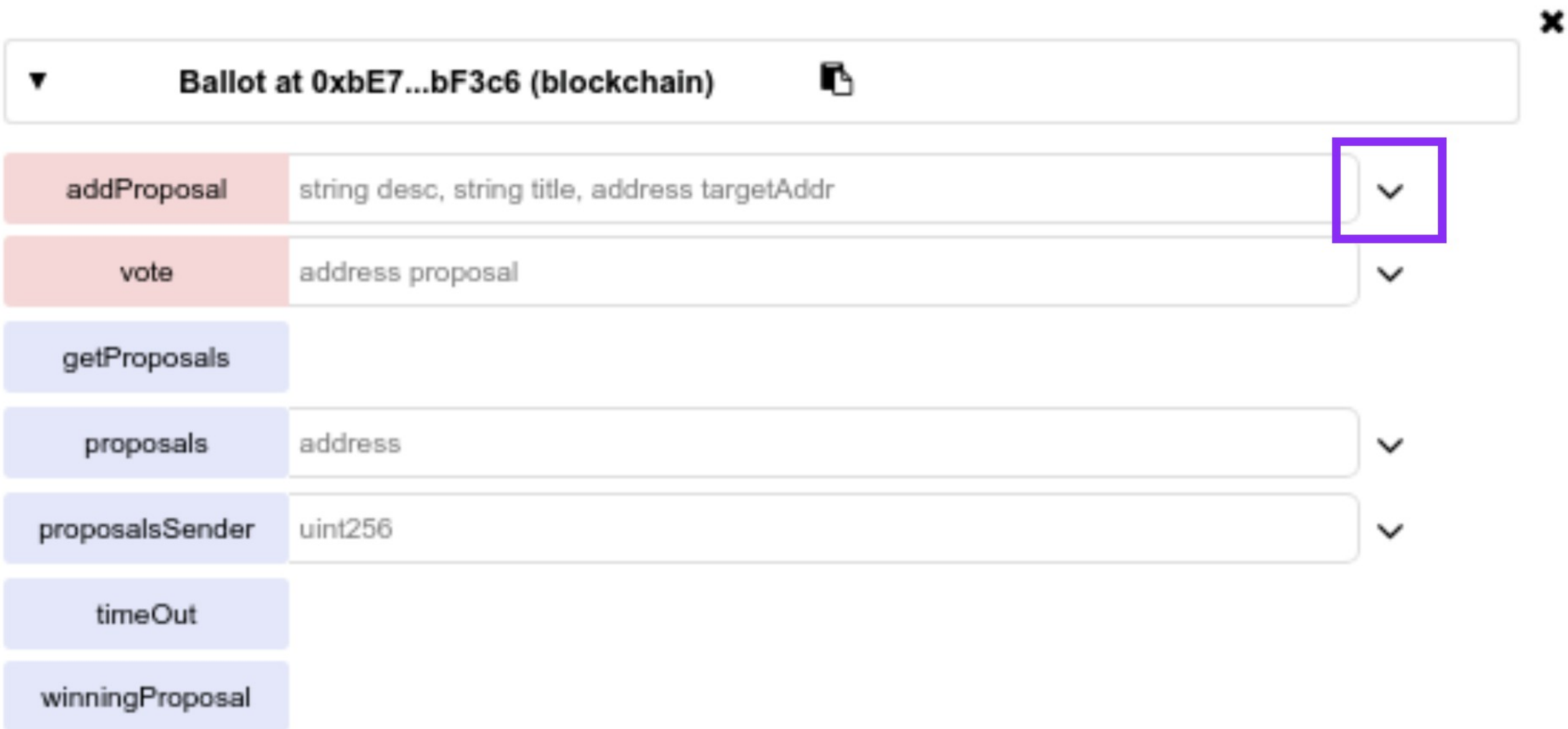


Part III

Adding a proposal and viewing the DAPP

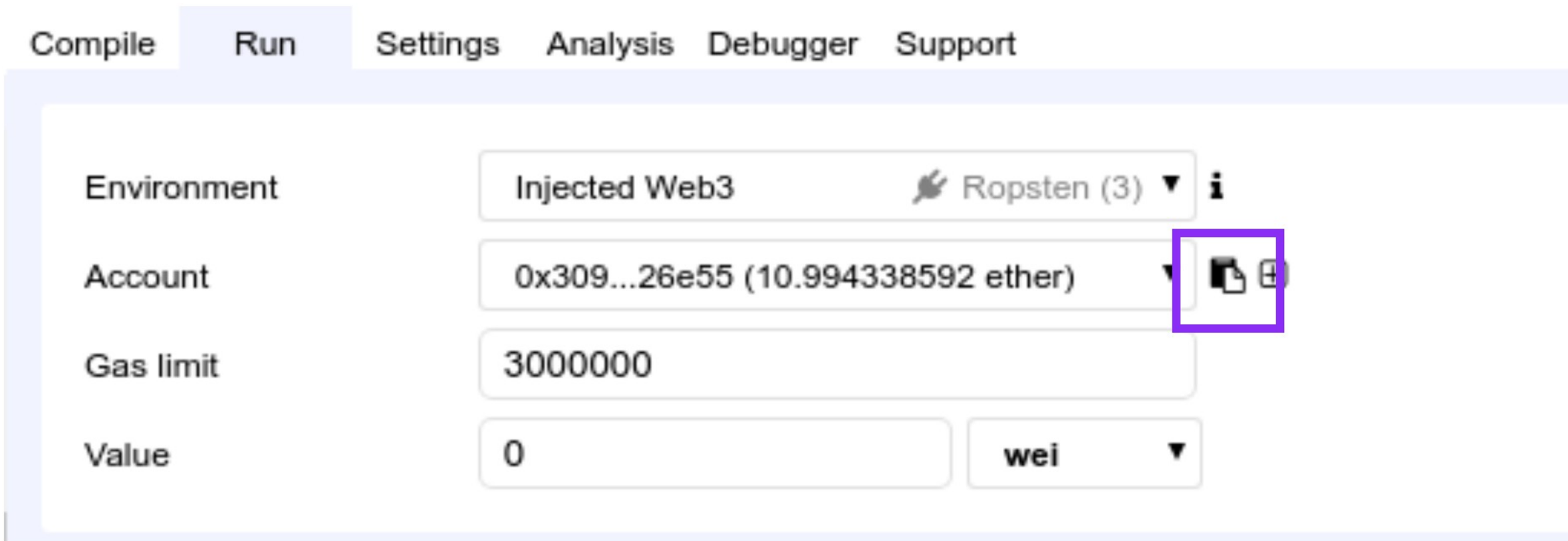
Add a new proposal

Expand addProposal function



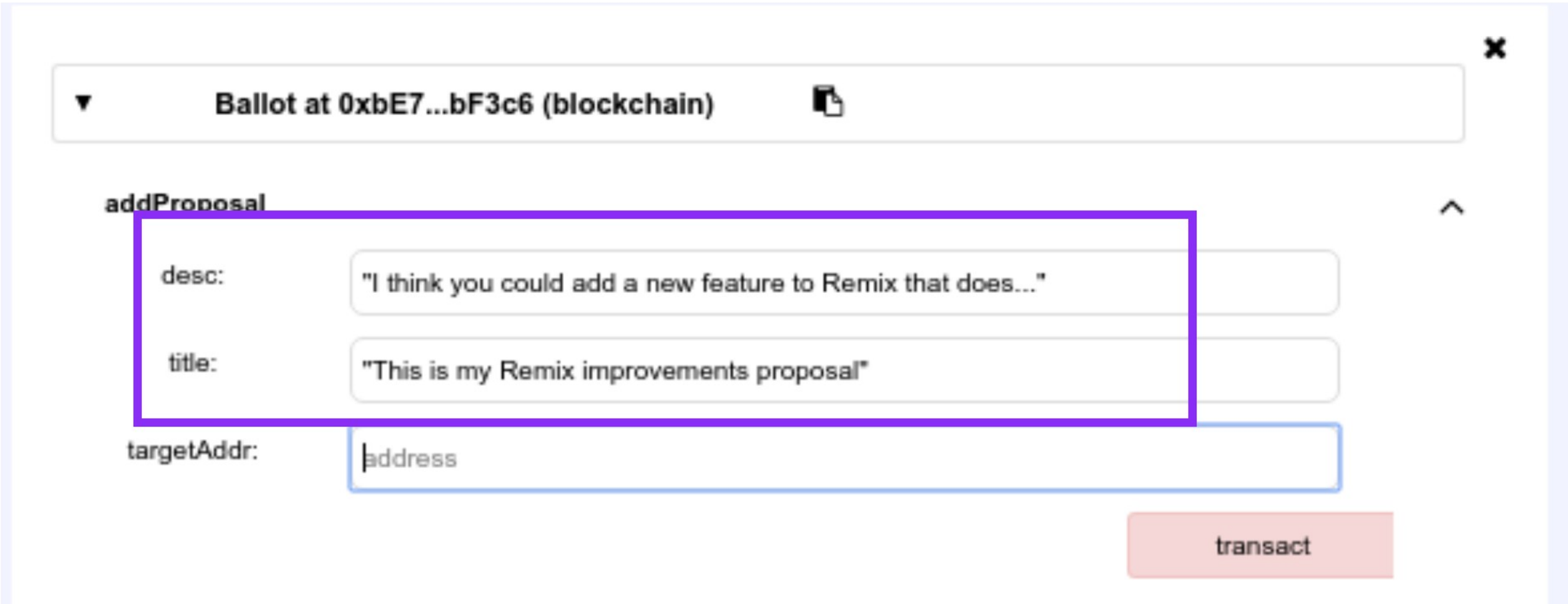
Copy your address

Run tab: Account



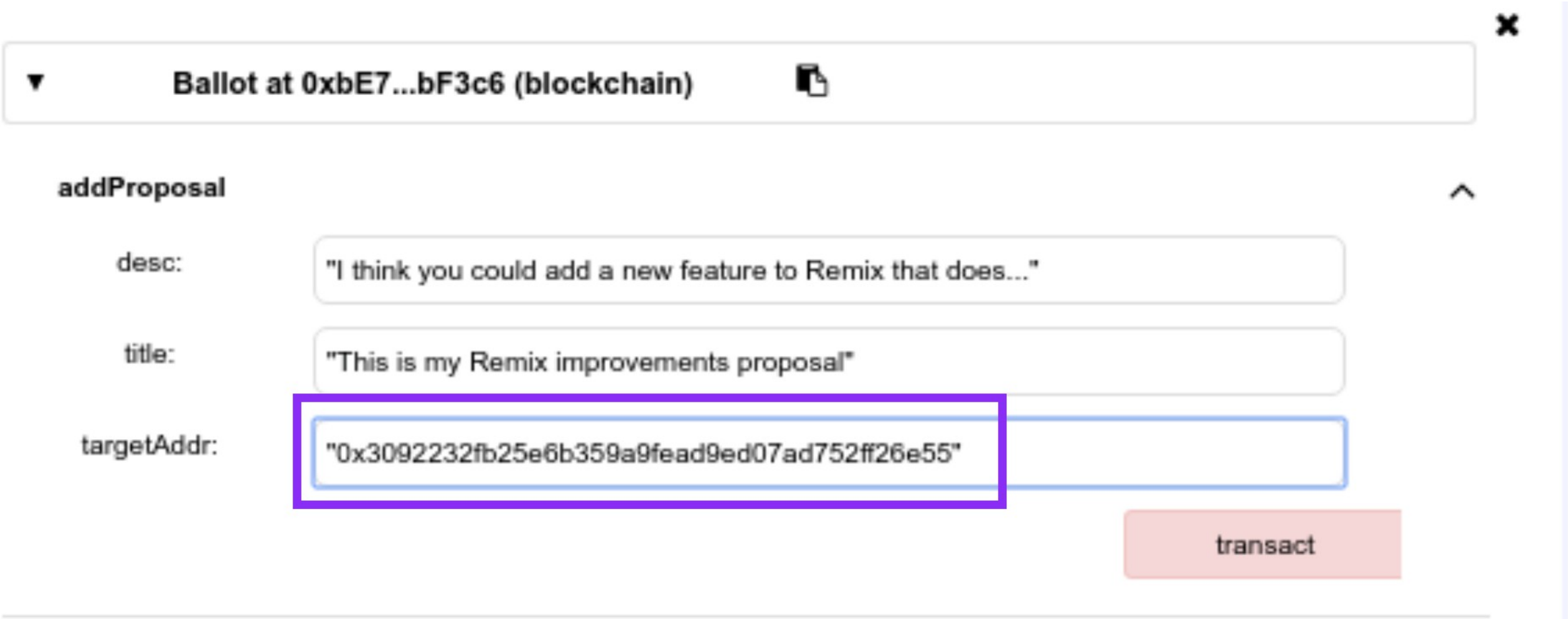
Type a proposal

Run tab: Account



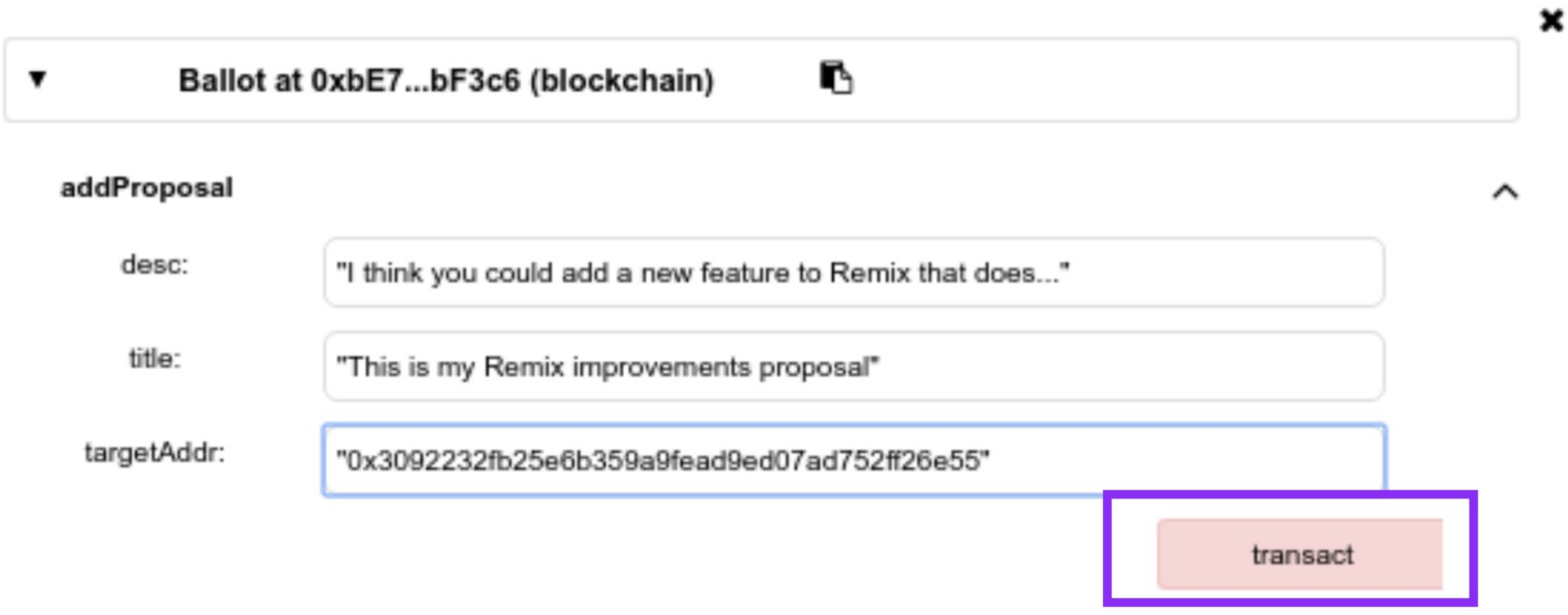
Add your address

Paste the address



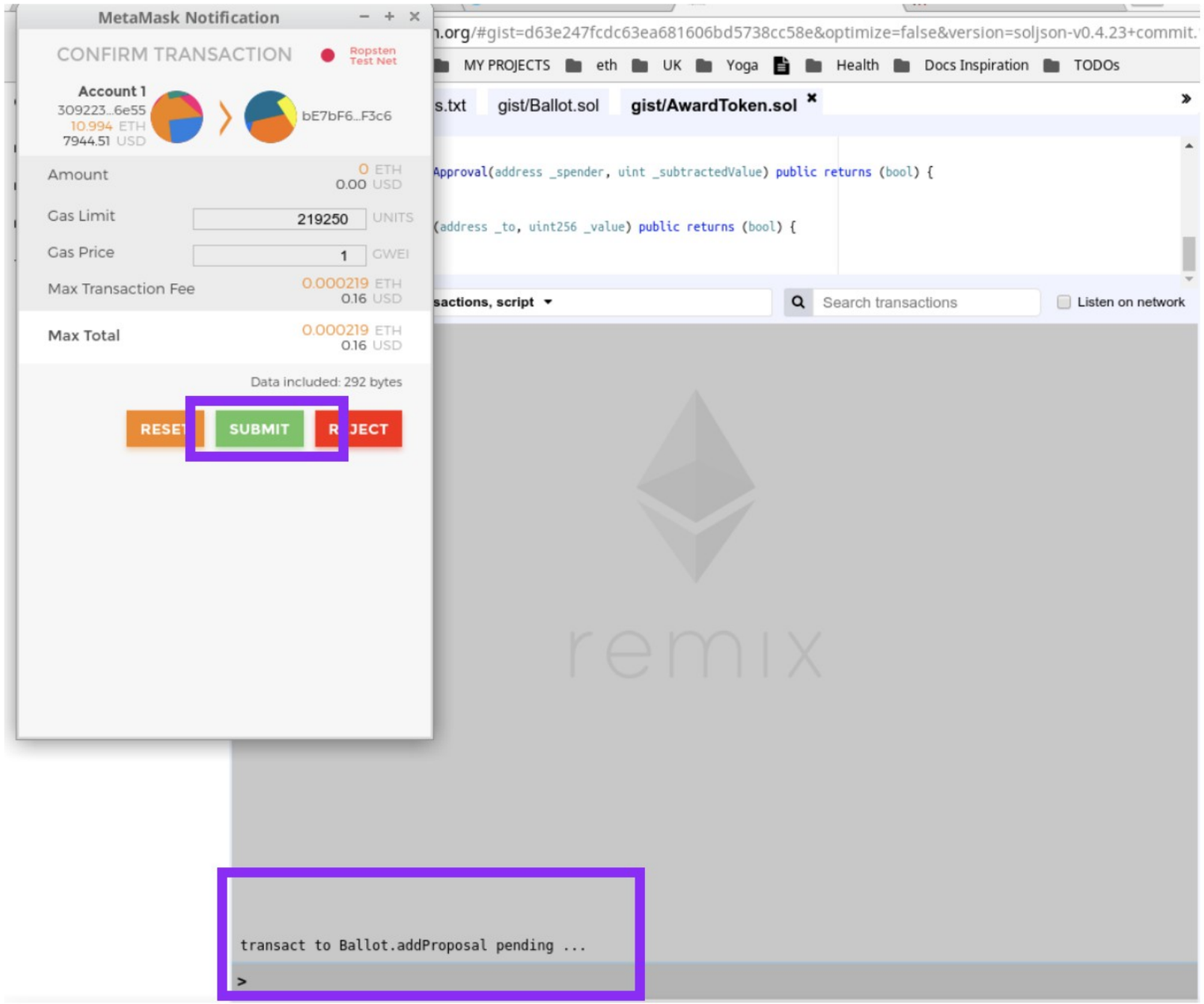
Execute addProposal

Hit the transact button



Confirm the transaction

Submit button



Check if tx succeeded

Terminal logs in Remix

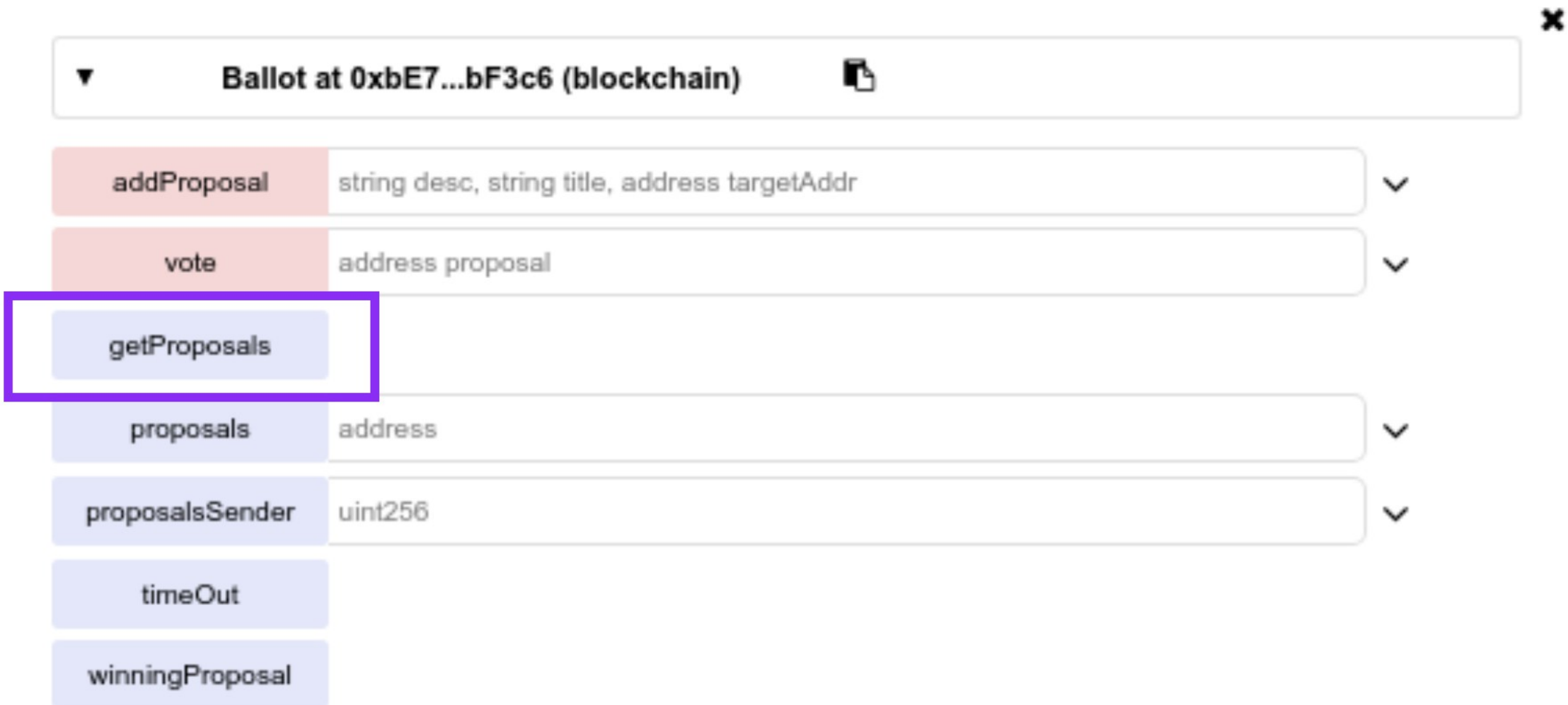
```
transact to Ballot.addProposal pending ...

https://ropsten.etherscan.io/tx/0x82017ad20f4a74978f46662863b60f364a79092ef3f6aaa4f0add9696891d3dd

[block:3159638 txIndex:8]  from:0x309...26e55
to:Ballot.addProposal(string,string,address) 0xbe7...bf3c6 value:0 wei data:0x54c...00000 logs:0
hash:0x820...1d3dd
```

Debug

status	0x1 Transaction mined and execution succeed
from	0x3092232fb25e6b359a9fead9ed07ad752ff26e55
to	Ballot.addProposal(string,string,address) 0xbe7bf60cee009dcdb2ad8d045c19e76597bbf3c6
gas	219250 gas
transaction cost	219250 gas
input	0x54c496e100000000000000000000000000000000000000000000000000000000000000006000000000000000000000000000000000000000000000000000000000000000c000000000000000000000000003092232fb25e6b359a9fead9ed07ad752ff26e550000000000000000000000000000000000000000000000000000000000

Execute getProposals

getProposals call



See Proposals Addresses

well in so far there will only be 1 address

call to Ballot.getProposals

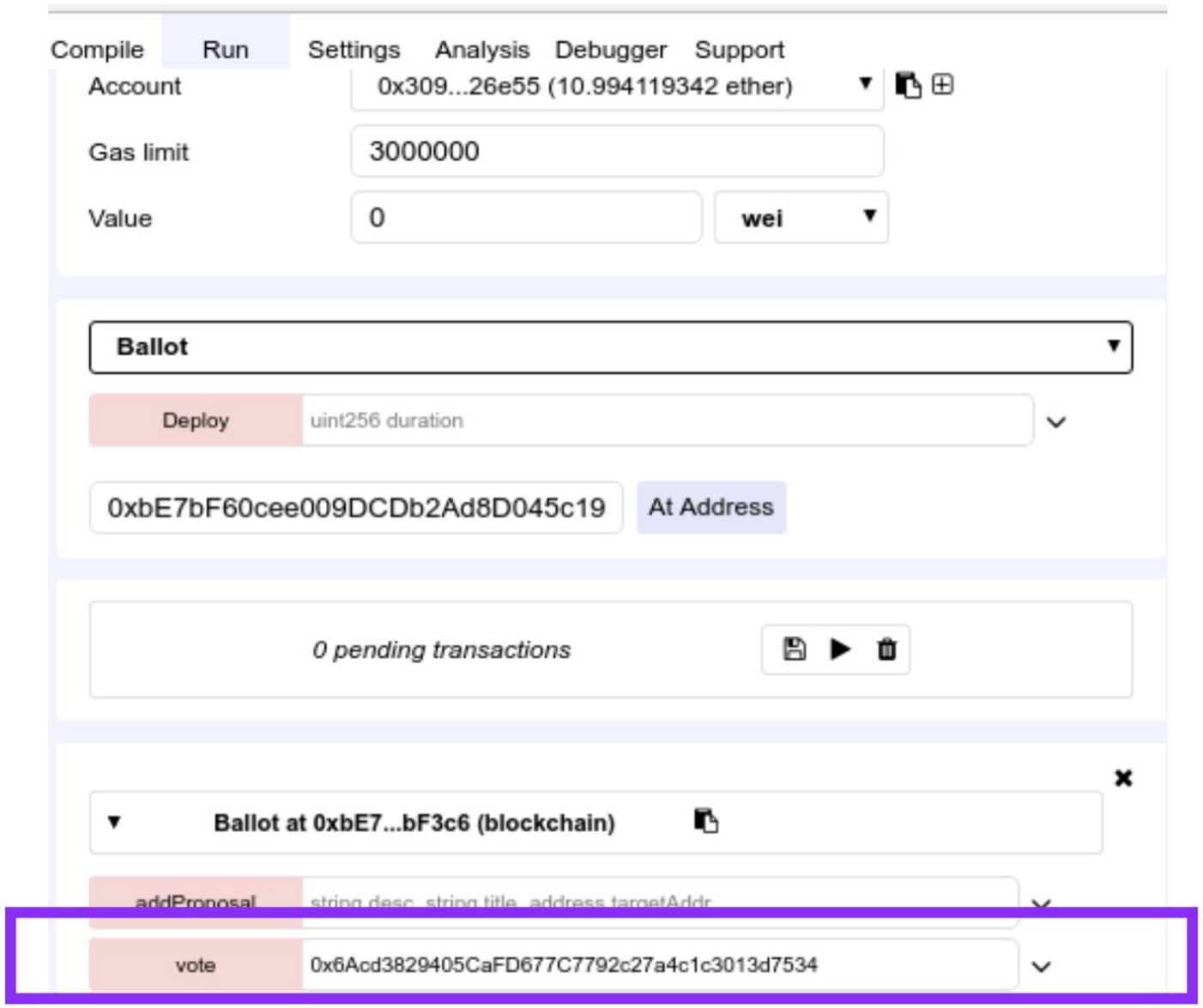
[call] **from:**0x3092232fb25e6b359a9fead9ed07ad752ff26e55 **to:**Ballot.getProposals()
data:0x625...64c48

Debug

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

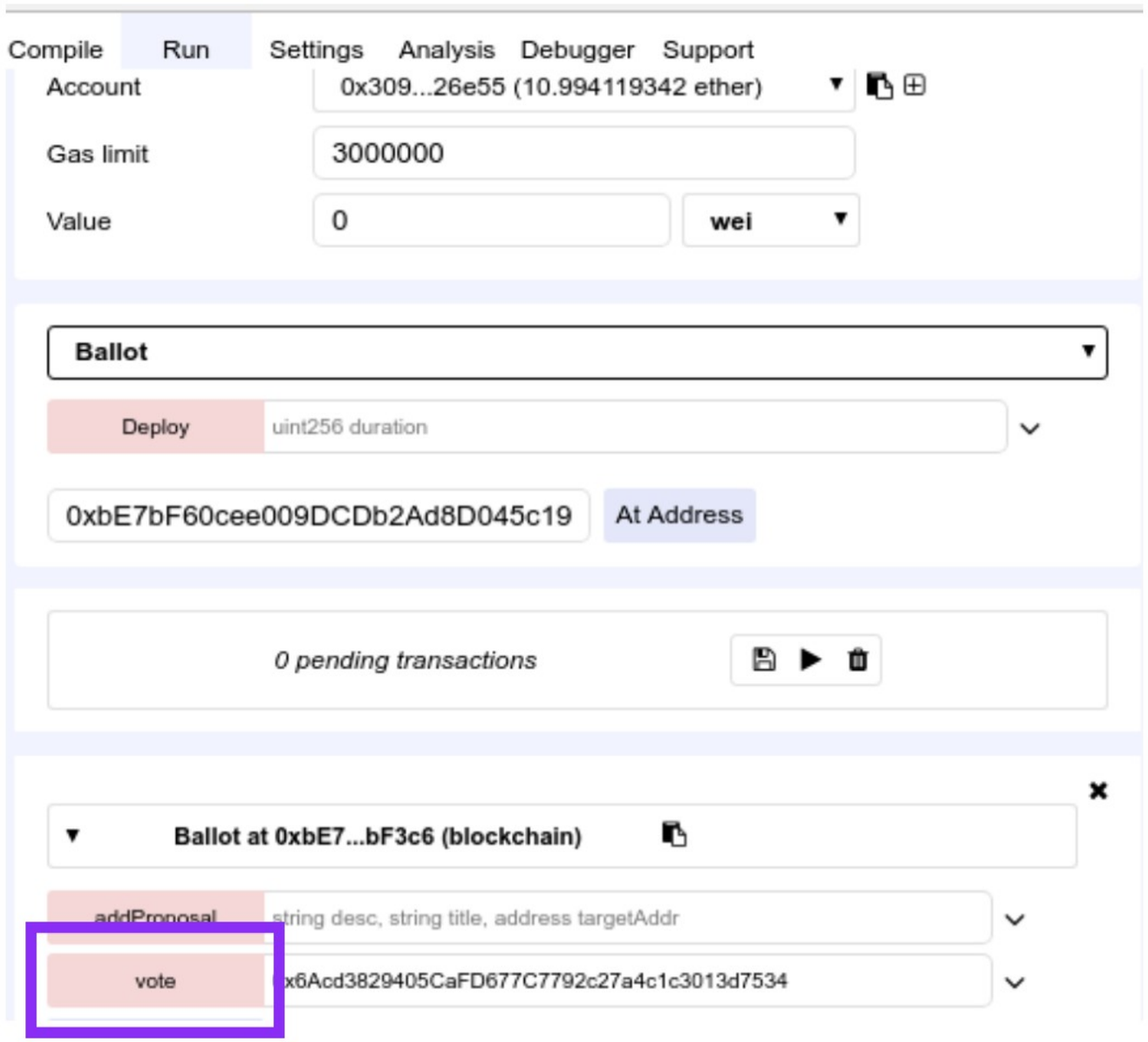
Vote for one Proposal

Paste Proposal Address you want to vote for



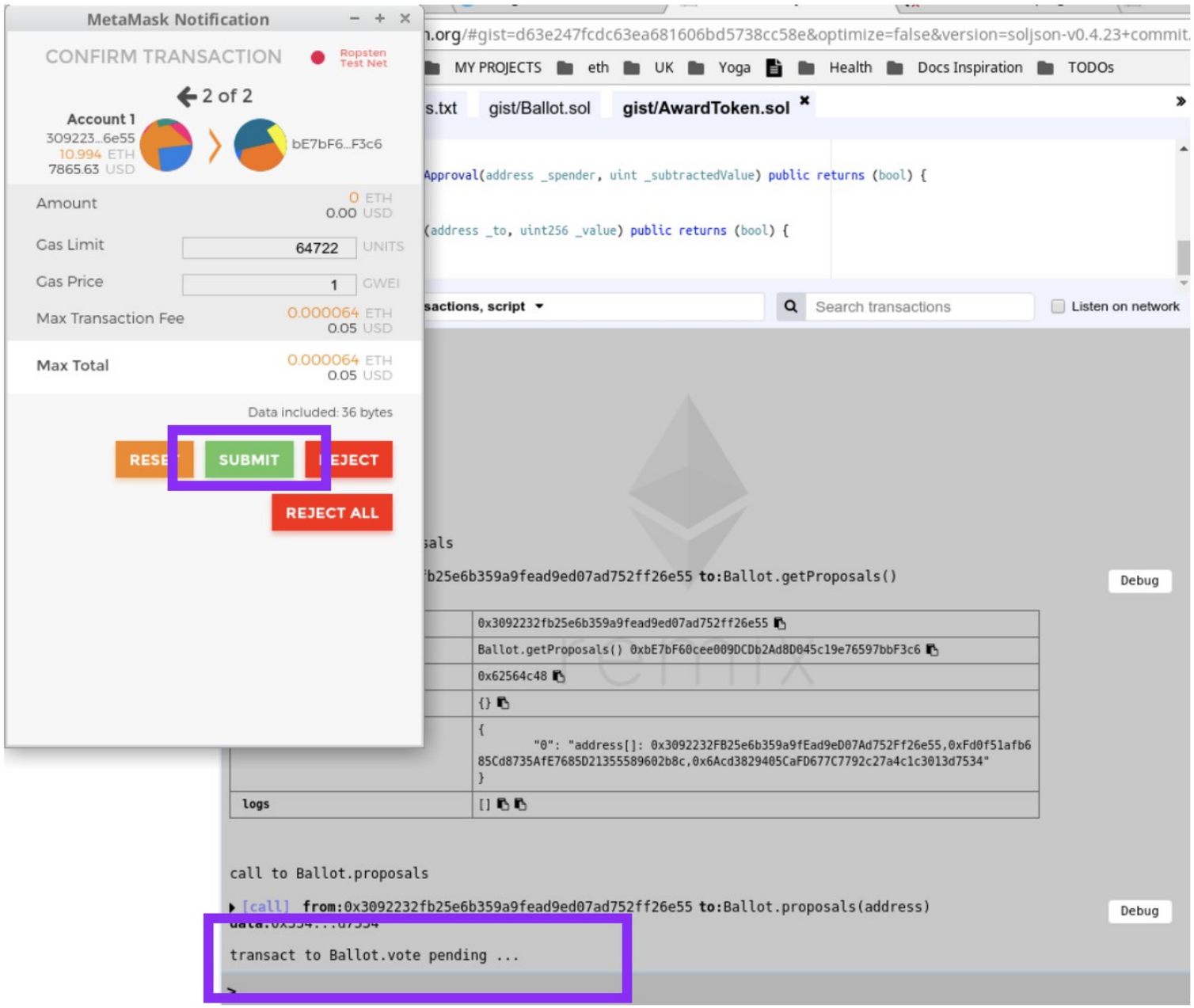
Execute vote transaction

vote button



Confirm the transaction

Submit button



Check if tx succeeded

Terminal logs in Remix

gist/dependencies.txt | gist/Ballot.sol | gist/AwardToken.sol

- browser
- config
- github
- gist
  - AwardToken.sol
  - Ballot.sol
  - TUTORIAL.md
  - dependencies.txt

```
63      }
64
65      function decreaseApproval(address _spender, uint _subtractedValue) public returns (bool) {
66          revert();
67      }
68
69      function transfer(address _to, uint256 _value) public returns (bool) {
70          revert();
71      }
72
73  }
```

[2] only remix transactions, script | Search transactions | Listen on network

```
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	[]

```
call to Ballot.proposals

[call] from:0x3092232fb25e6b359a9fead9ed07ad752ff26e55 to:Ballot.proposals(address)
data:0x334...d7534

transact to Ballot.vote pending ...

[block:3159861 txIndex:27] from:0x309...26e55 to:Ballot.vote(address) 0xbe7...bf3c6 value:0 wei
data:0x6dd...d7534 logs:0 hash:0xe0d...6c6eb
```

Debug

Debug

Now let's try it out connecting a frontend

http://bit.ly/remix-voting

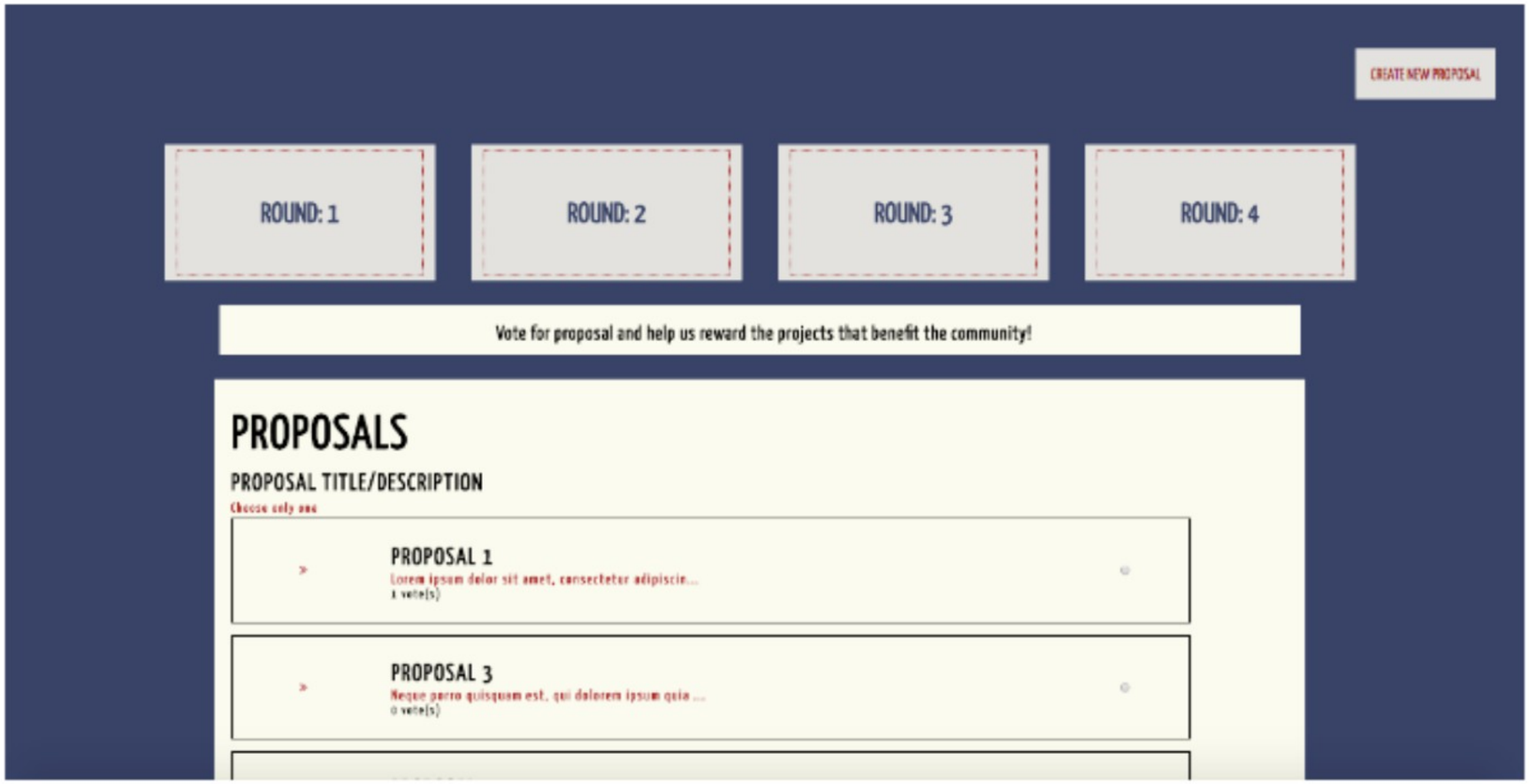
(when dependencies.js is the active file)

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;
```

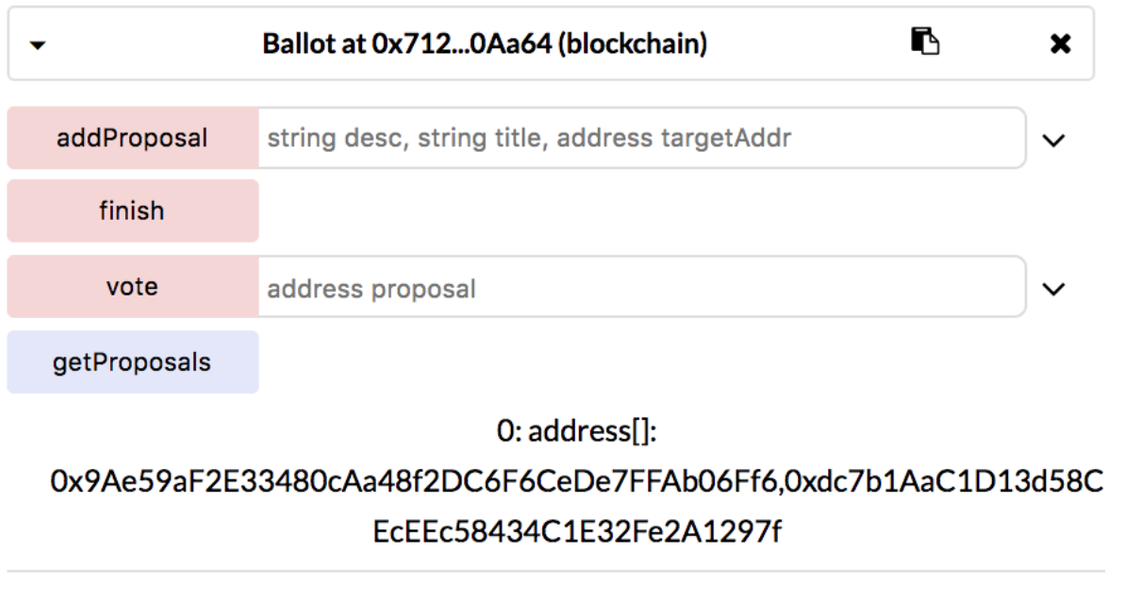
```
contract AwardToken is MintableToken {
    uint quantity;
    uint ballotPeriod = 7 hours;
    Ballot public currBallot;
    address[] public prevWinners;
```



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 @LianaHus @iurimatias

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