

Building Smart Contracts with Remix

NINA BREZNIK

@ninabreznik

YANN LEVREAU

@ninabreznik

IURI MATIAS

@jurimatias

ROB STUPAY

@ryestew

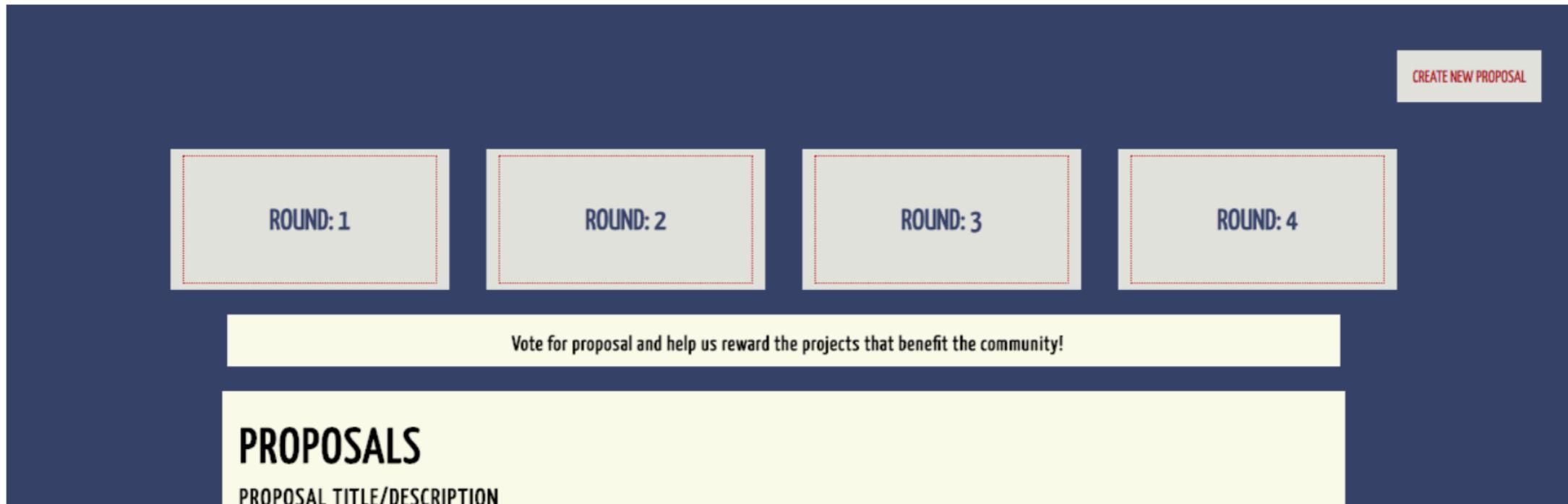
ALEX PRAETORIUS

@SERAPATH

Ballot Dapp Workshop

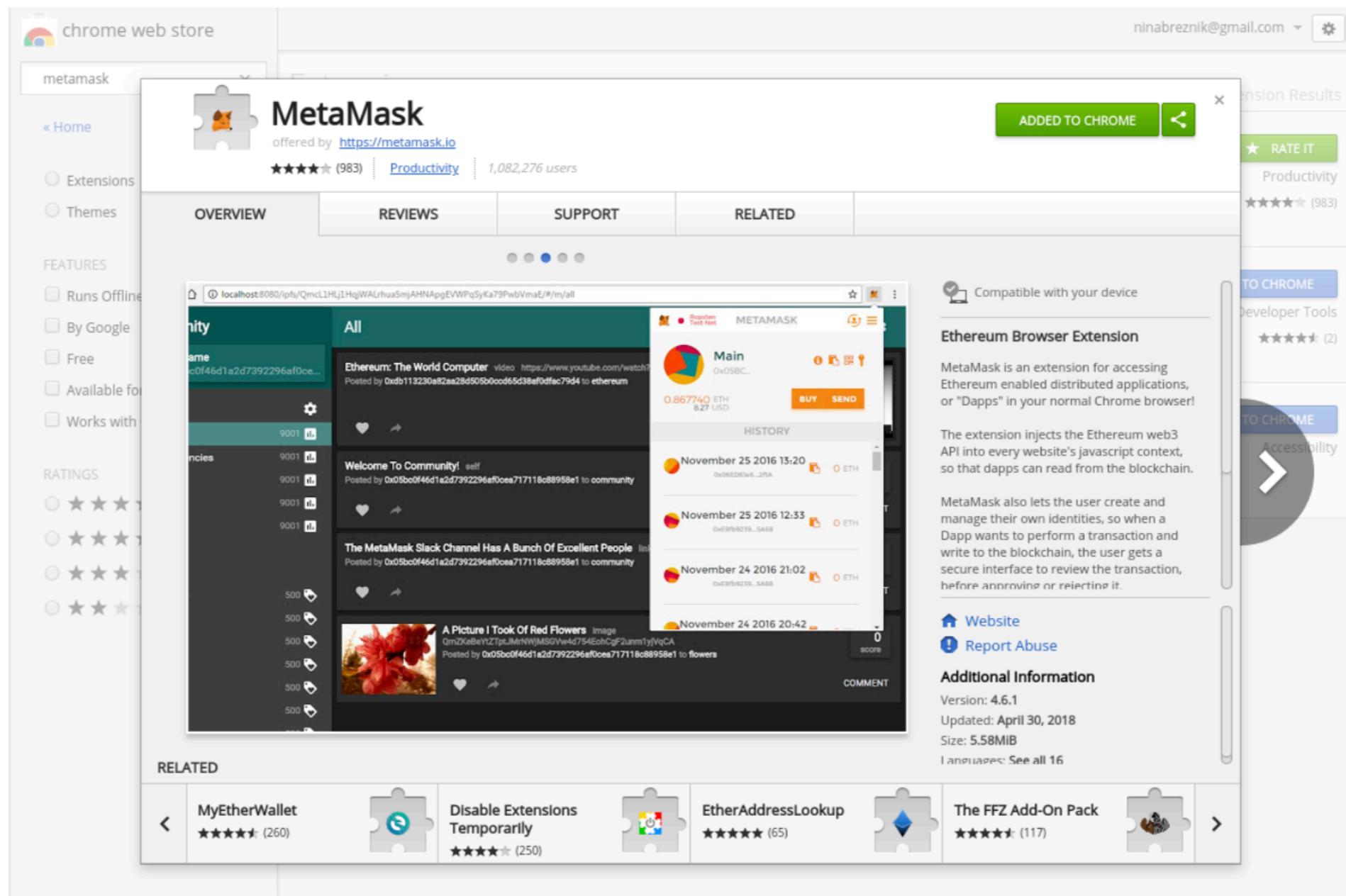
bit.ly/remix-workshop-repository

PDF: <https://updig.is/pdf/remix-chez-coinhause.pdf>



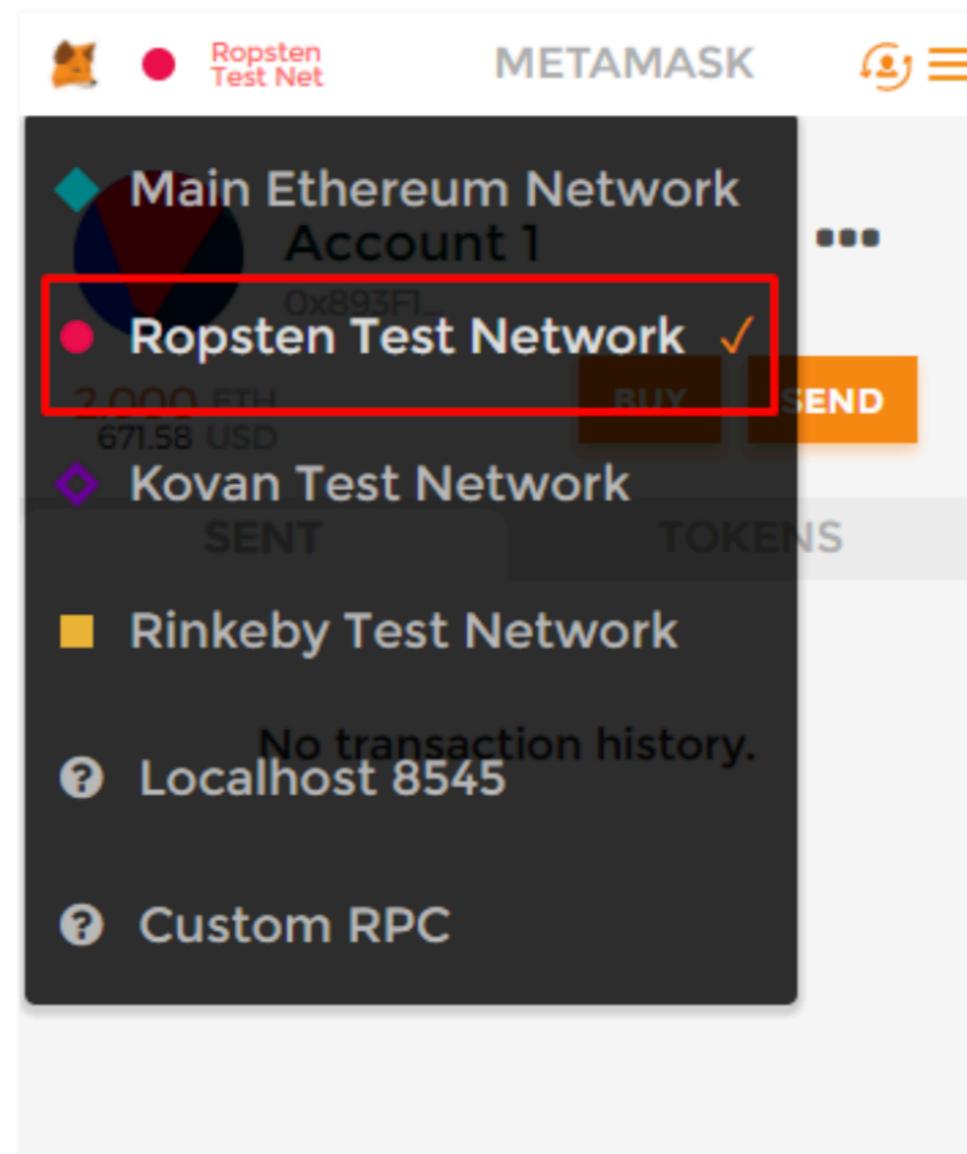
Install Metamask

chrome.google.com/webstore



Login to Metamask

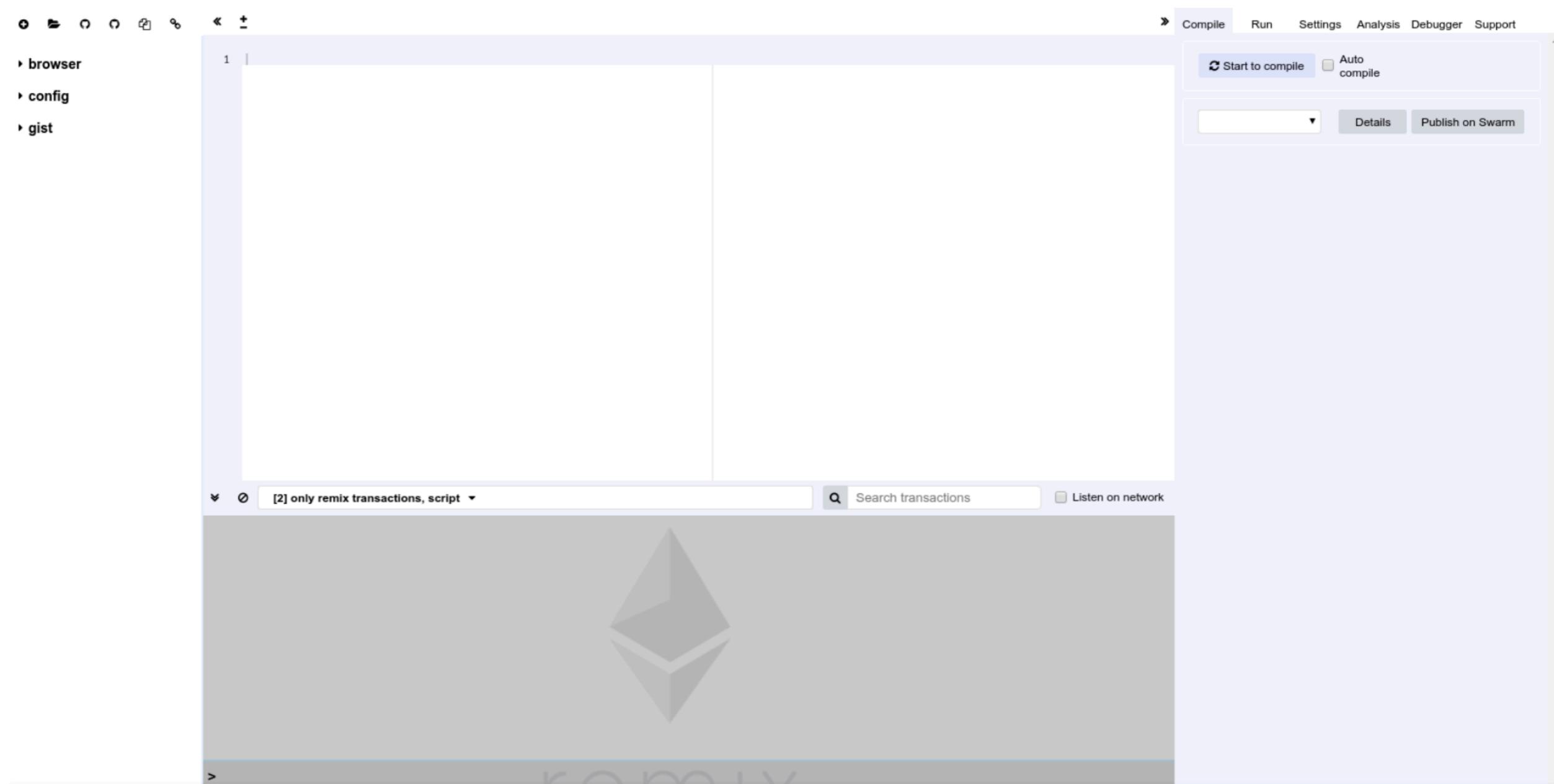
Robsten Test Network



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)

The screenshot shows the Remix IDE interface with several tabs open:

- File Explorer:** Shows a sidebar with project files: browser/AwardToken.sol, browser/Ballot2.sol, browser/Ballot_orig.sol (selected), browser/Donation.sol, README.md, multiSig2.sol, multisig.sol, multisig1.sol, scenario.json, setup.txt, and config.
- Editor:** Displays Solidity code for a Ballot contract. The code defines a Voter struct with fields weight, voted, vote, and delegate; a Proposal struct with field voteCount; and variables chairperson, voters, and proposals. A warning is shown at line 19: "browser/Ballot_orig.sol:19:5: Warning: Defining function Ballot(uint8 _numProposals) public ^ (Relevant source part starts here and spans multiple lines)".
- Terminal:** Shows the welcome message for Remix v0.6.4 and a list of available commands:

 - 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 environment is injected provider 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 gist, swarm or ipfs.
 - remix.setproviderurl(url): Change the current provider to Web3 provider and set the url endpoint.
 - remix.exeCurrent(): Run the script currently displayed in the editor
 - remix.help(): Display this help message

- Console:** Shows a blank console area.

Run Tab

Compile Run Settings Analysis Debugger Support Test

The screenshot shows the 'Run' tab interface of a blockchain development tool. At the top, there are tabs for Compile, Run, Settings, Analysis, Debugger, Support, and Test. The Run tab is active.

Environment: Set to Injected Web3, Ropsten (3) network.

Account: Address 0x9ae...06ff6 (1.992485469305616838) with a file icon and a plus sign.

Gas limit: 3000000.

Value: 0 wei.

Contract Selection: A dropdown menu shows 'AwardToken'.

Deployment Options: A pink 'Deploy' button, a 'Load contract from Address' input field, and an 'At Address' button.

Transactions Recorded: 4 transactions recorded.

Deployed Contracts: A list of deployed contracts, with a delete icon.

- 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>

The screenshot shows the Remix IDE interface. On the left, the file tree under 'browser' includes AwardToken.sol, Ballot2.sol, Ballot_orig.sol, Donation.sol, README.md, multiSig2.sol, multisig.sol, multisig1.sol, scenario.json, and setup.txt. The 'Ballot_orig.sol' file is open in the center editor, displaying Solidity code for a Ballot contract. The right panel contains a 'Compiler' tab bar with 'Compile', 'Run', 'Settings', 'Analysis', 'Debugger', and 'Support'. A 'Start to compile' button and an 'Auto compile' checkbox are visible. Below the compiler tabs is a 'Ballot' section with 'Details', 'Publish on Swarm', 'ABI', and 'Bytecode' buttons. A purple box at the bottom right indicates 'Static Analysis raised 2 warning(s) that requires your attention.' A yellow box highlights a warning message: 'browser/Ballot_orig.sol:19:5: Warning: Defining function Ballot(uint8 _numProposals) public ^ (Relevant source part starts here and spans...'. At the bottom, a terminal window displays the welcome message for Remix v0.6.4 and a list of commands. A red box highlights the command documentation section.

```
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;
}
```

- 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 environment is injected provider 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 git, hub, swarm or ipfs.

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

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

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

Set environment

Run tab: Environment = Injected web3
(Ropsten)

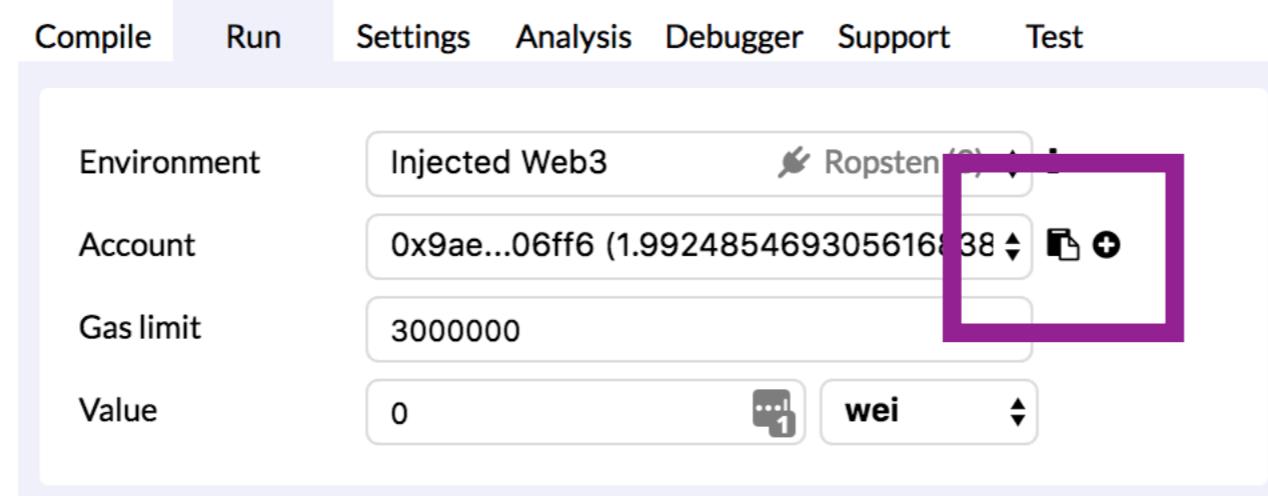
```
ContractDefinition AwardToken 0 reference(s) ▾
1 import "github/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/MintableToken.sol";
2 import "gist/Ballot.sol";
3
4 contract AwardToken is MintableToken {
5     uint quantity;
6     uint ballotPeriod = 7 hours;
7     Ballot public currBallot;
8     address[] public prevWinners;
9     event log (string _msg);
10    event winLog (address _win);
11    event newBallot (address _addr);
12
13    function AwardToken () {
14        quantity = 100;
15    }
16
17    function getPreviousWinners() constant returns (address[]) {
18        return prevWinners;
19    }
20
21    // either a name change or it works fine without it
22    // function approve(address spender, uint256 value) public returns (bool);
23    function startRound() onlyOwner canMint public returns (bool) {
24        // if this is the first minting then we should let this go immediately
25    }
```

Get some TEST ether

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

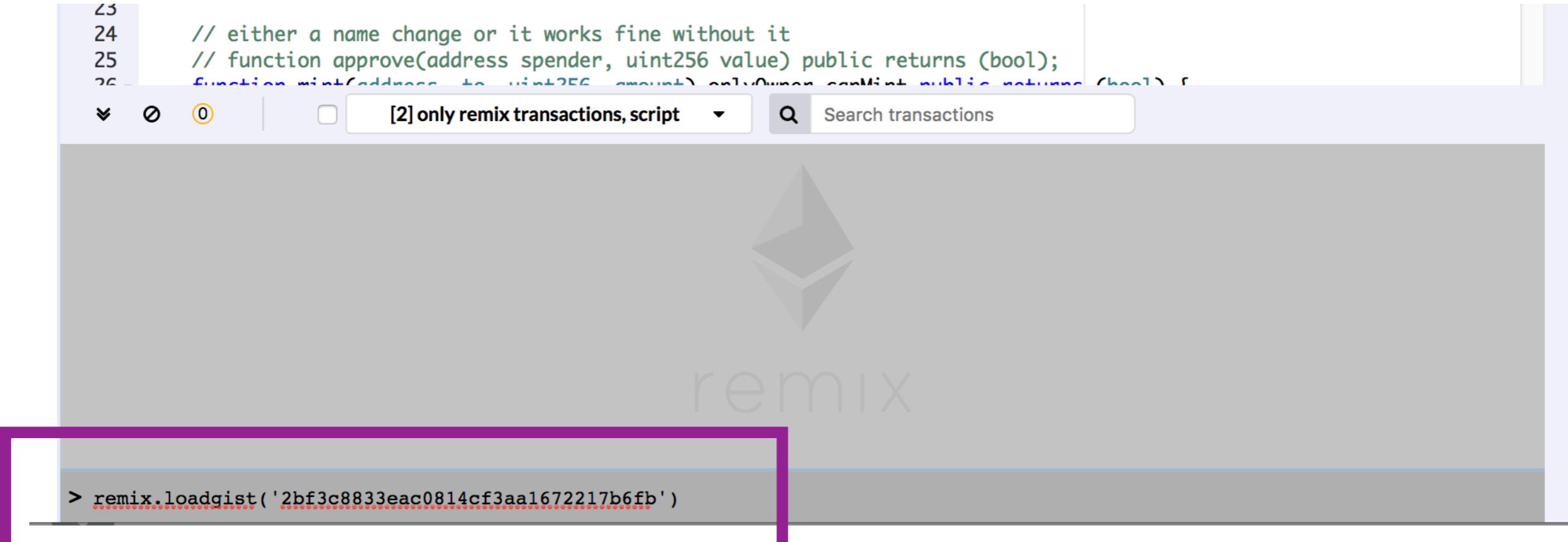
But FIRST:

Copy your address - here or in Metamask



Load files to Remix

```
remix.loadgist('add9633c2b0101f6fda0aadcfe350f60')
```



The screenshot shows the Remix IDE interface. At the top, there is a code editor window displaying Solidity code with line numbers 23 through 26. Below the code editor is a toolbar with various icons. To the right of the toolbar is a dropdown menu set to '[2] only remix transactions, script' and a search bar labeled 'Search transactions'. The main workspace below the toolbar features the Ethereum logo and the word 'remix'. In the bottom left corner of the workspace, there is a purple rectangular box containing a command-line interface (CLI) window. The CLI window has a light gray background and displays the command: > `remix.loadgist('2bf3c8833eac0814cf3aa1672217b6fb')`. The entire CLI window is highlighted with a thick purple border.

here in the console

Compile AwardToken

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

Click the “Start to compile” button

Update image w/o github

The screenshot shows the Truffle UI interface. On the left, there is a sidebar with a tree view of project files. The 'gist' section contains 'AwardToken.sol'. The main area displays the Solidity code for 'AwardToken.sol'. The right side features a toolbar with tabs for 'Compile', 'Run', 'Analysis', 'Testing', 'Debugger', 'Settings', and 'Support'. Below the toolbar, a panel shows compiler settings: 'Current version: 0.4.25+commit.59dbf8f1.Emscripten clang', 'Select new compiler version', 'Auto compile' (checked), 'Enable Optimization' (unchecked), and 'Hide warnings' (unchecked). A prominent purple box highlights the 'Start to compile' button. At the bottom, there is a message about static analysis warnings and a snippet of warning text from 'Ballot.sol'.

```
1 import "github/OpenZeppelin/openzeppelin-zos/contracts/token/ERC20/MintableToken.sol";
2 import "gist/Ballot.sol";
3
4 contract AwardToken is MintableToken {
5     uint quantity;
6     uint ballotPeriod = 7 hours;
7     Ballot public currBallot;
8     address[] public prevWinners;
9     event log (string _msg);
10    event winLog (address _win);
11    event newBallot (address _addr);
12
13 function AwardToken () {
14     quantity = 100;
15 }
16
17 function getPreviousWinners() constant returns (address[])
18 {
19     return prevWinners;
20 }
21
22 // either a name change or it works fine without it
23 // function approve(address spender, uint256 value) public returns (bool)
24 // if this is the first minting then we should let this go immediately
25 function startRound() onlyOwner canMint public returns (bool) {
26     // if this is the first minting then we should let this go immediately
27     if (address(currBallot) == 0x0) {
        currBallot = new Ballot(ballotPeriod);
        newBallot(currBallot);
    }
}
```

Current version: 0.4.25+commit.59dbf8f1.Emscripten clang

Select new compiler version

Auto compile Enable Optimization Hide warnings

Start to compile

AwardToken

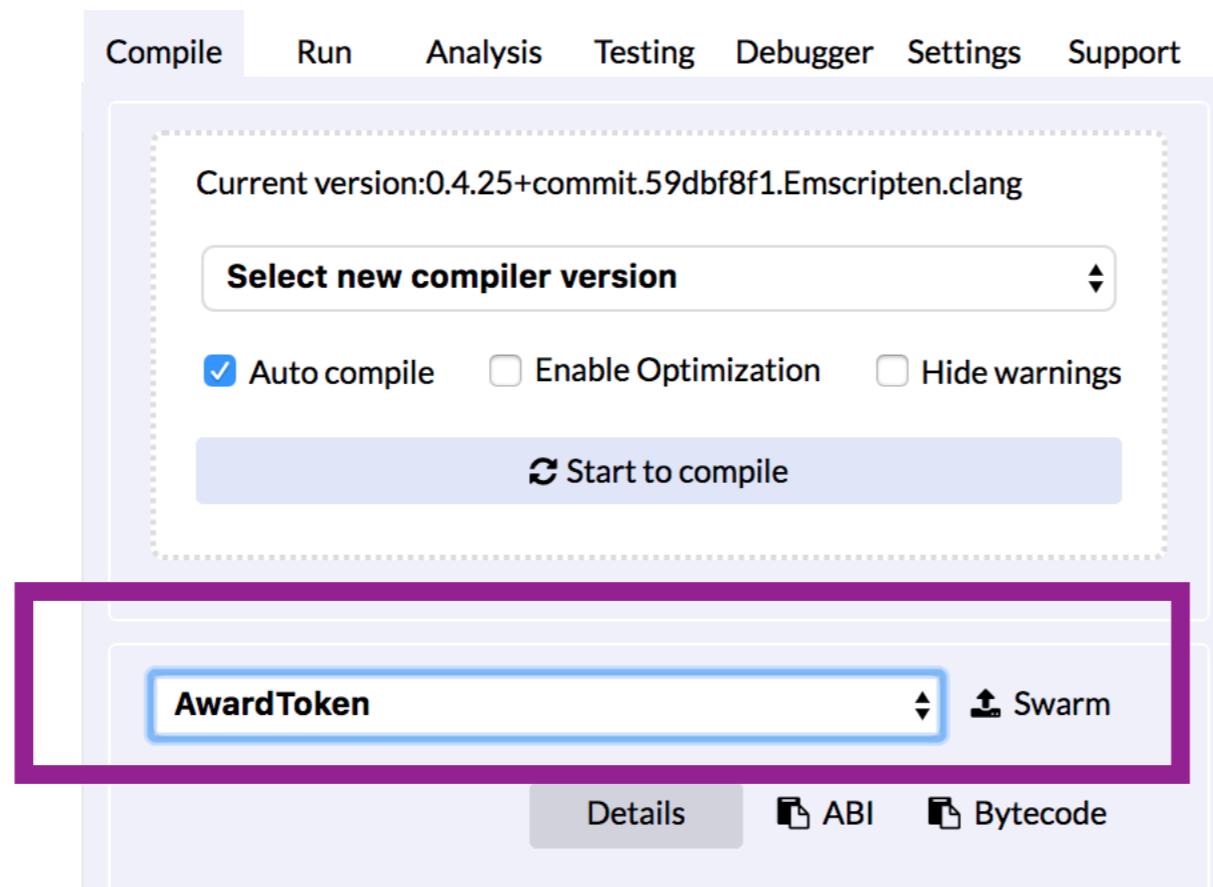
Static Analysis raised 52 warning(s) that requires your attention. Click here to show the warning(s).

gist/Ballot.sol:26:5: Warning: Defining constructors

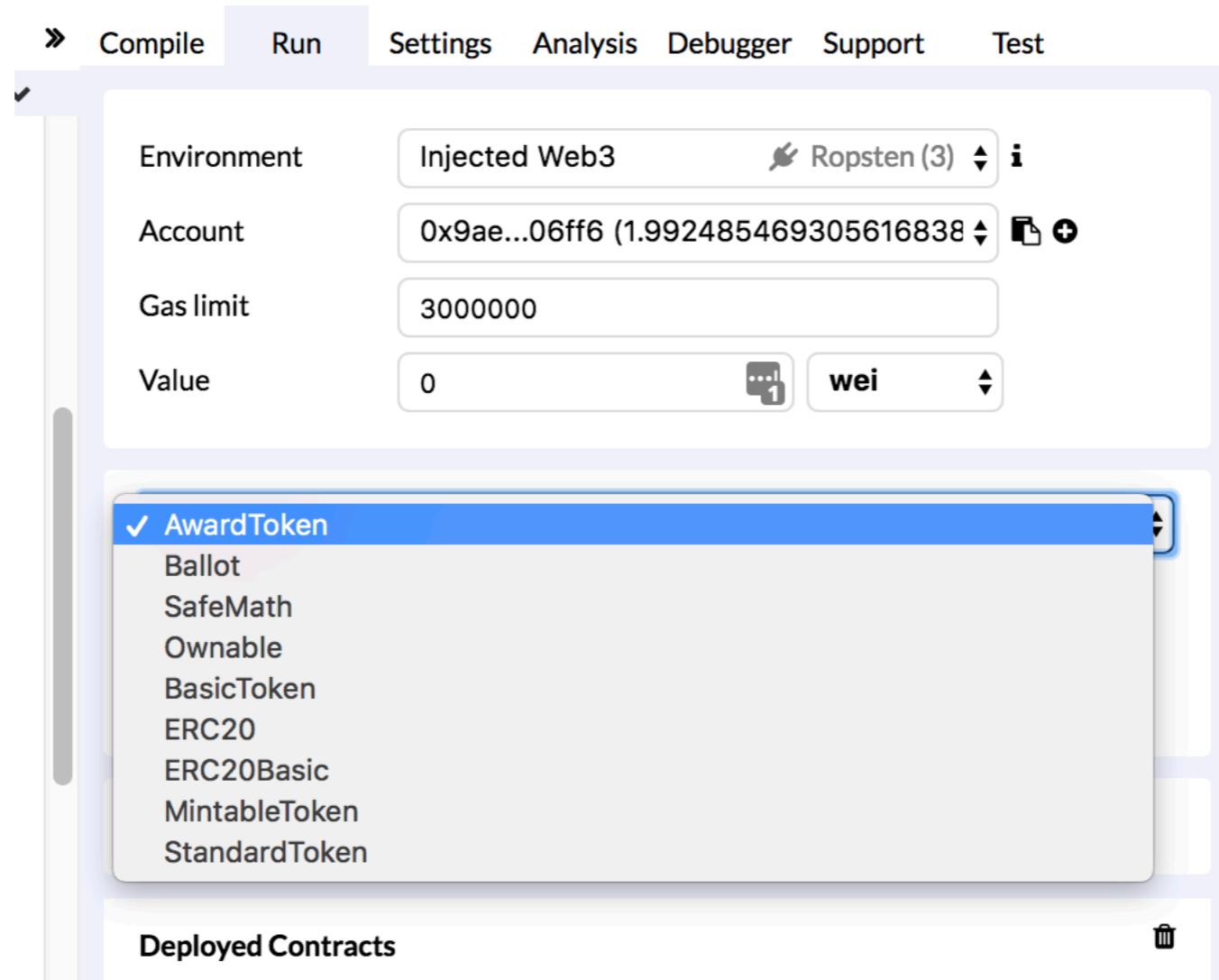
function Ballot(uint duration) public {
 ^ (Relevant source part starts here and spans across multiple lines)

See compiled contracts

AwardToken + all it's imported contracts

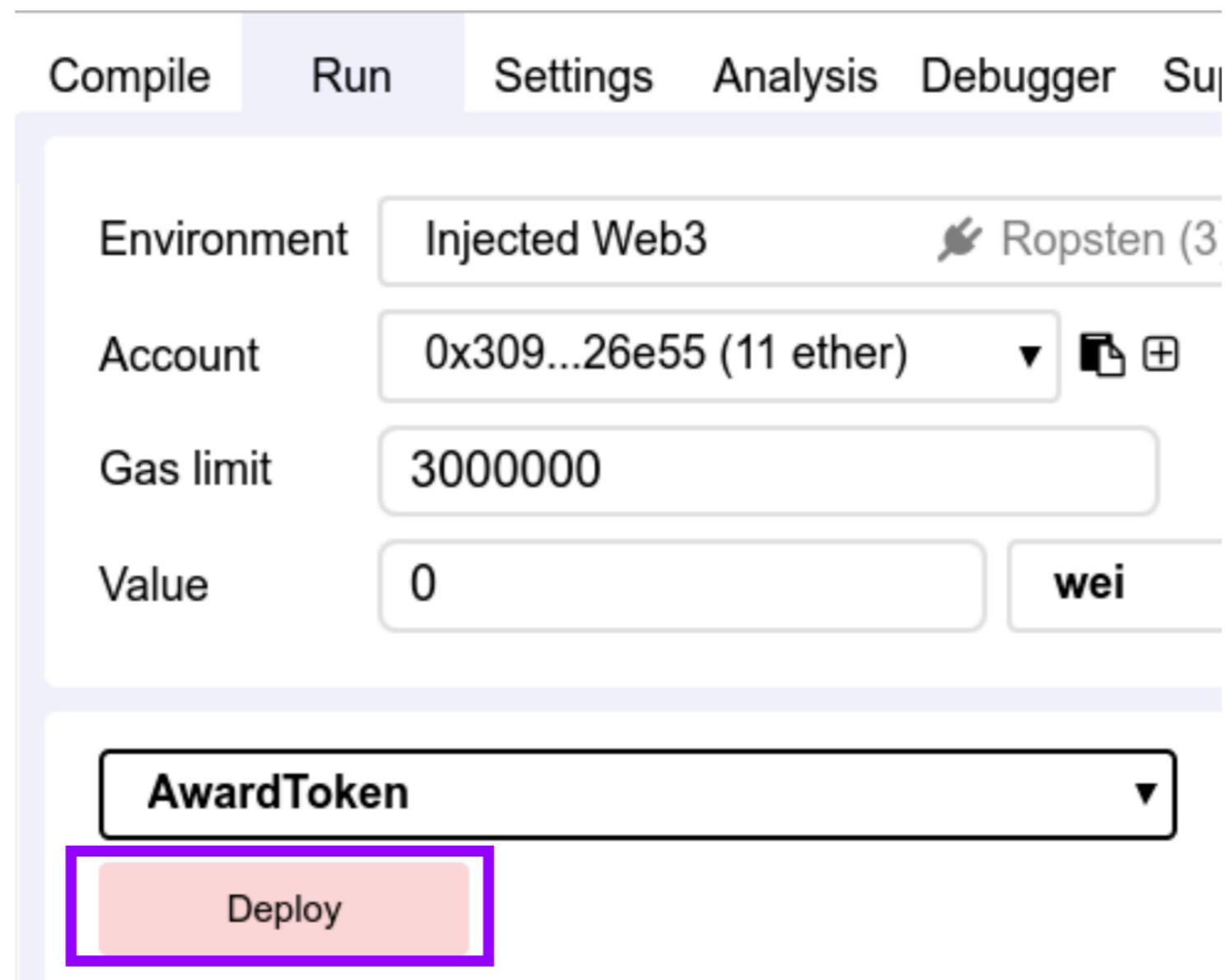


Imported Contracts



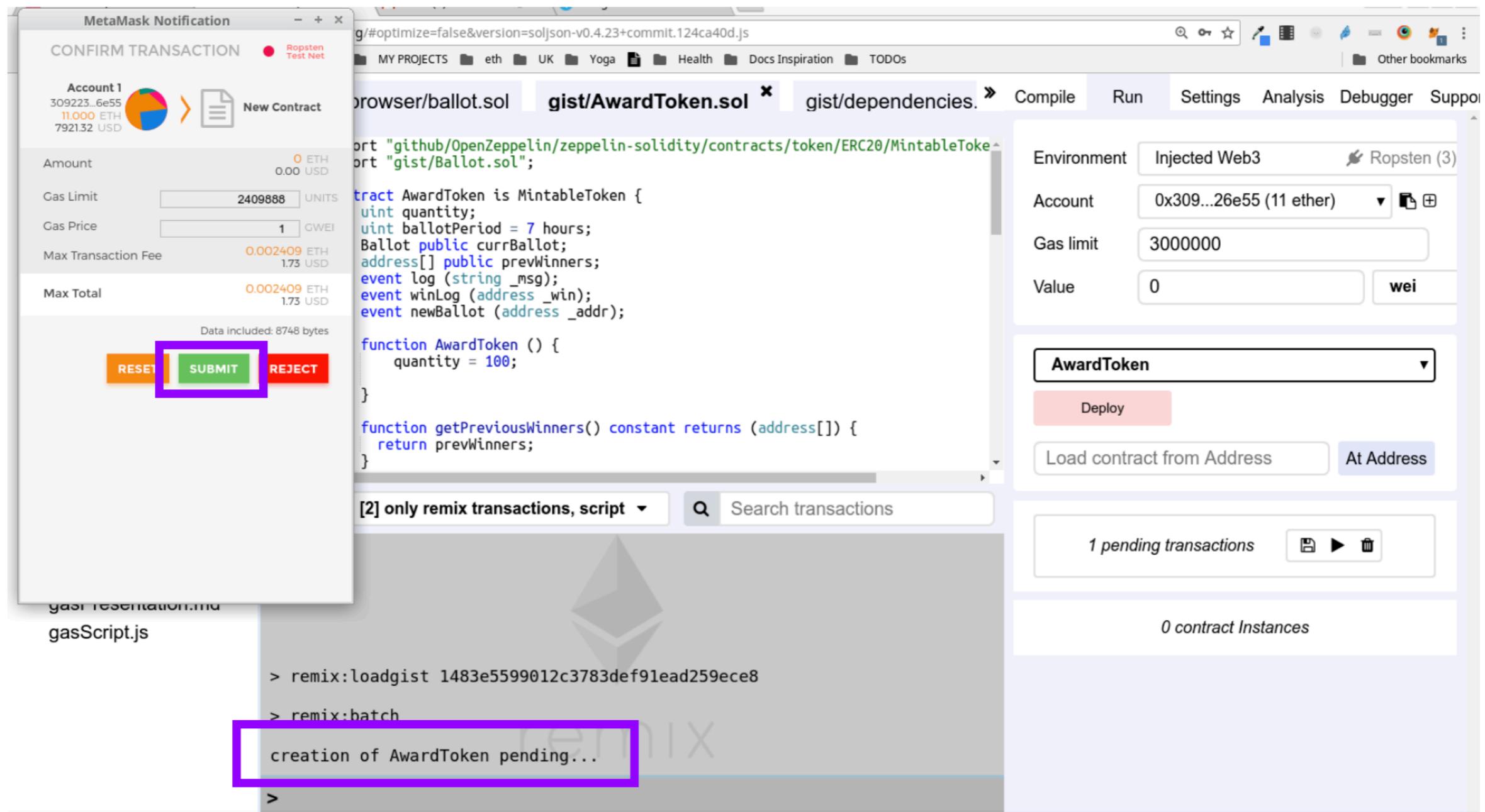
Deploy the contract

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...

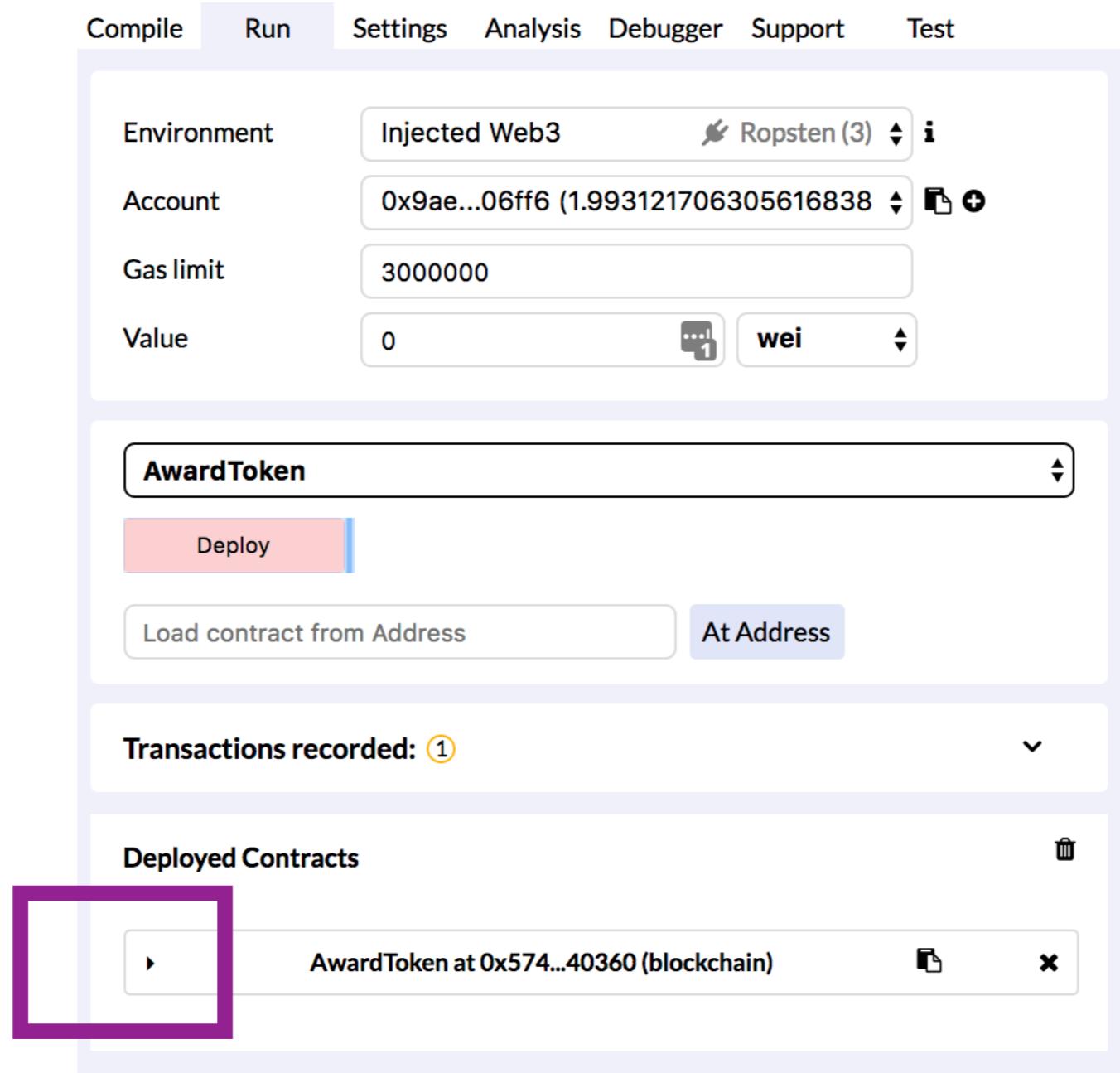
<https://ropsten.etherscan.io/tx/0x404a4445ebb3a969b15257a586a61582afa07dcf02b1b2617f77519b30378be8>

▶ [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

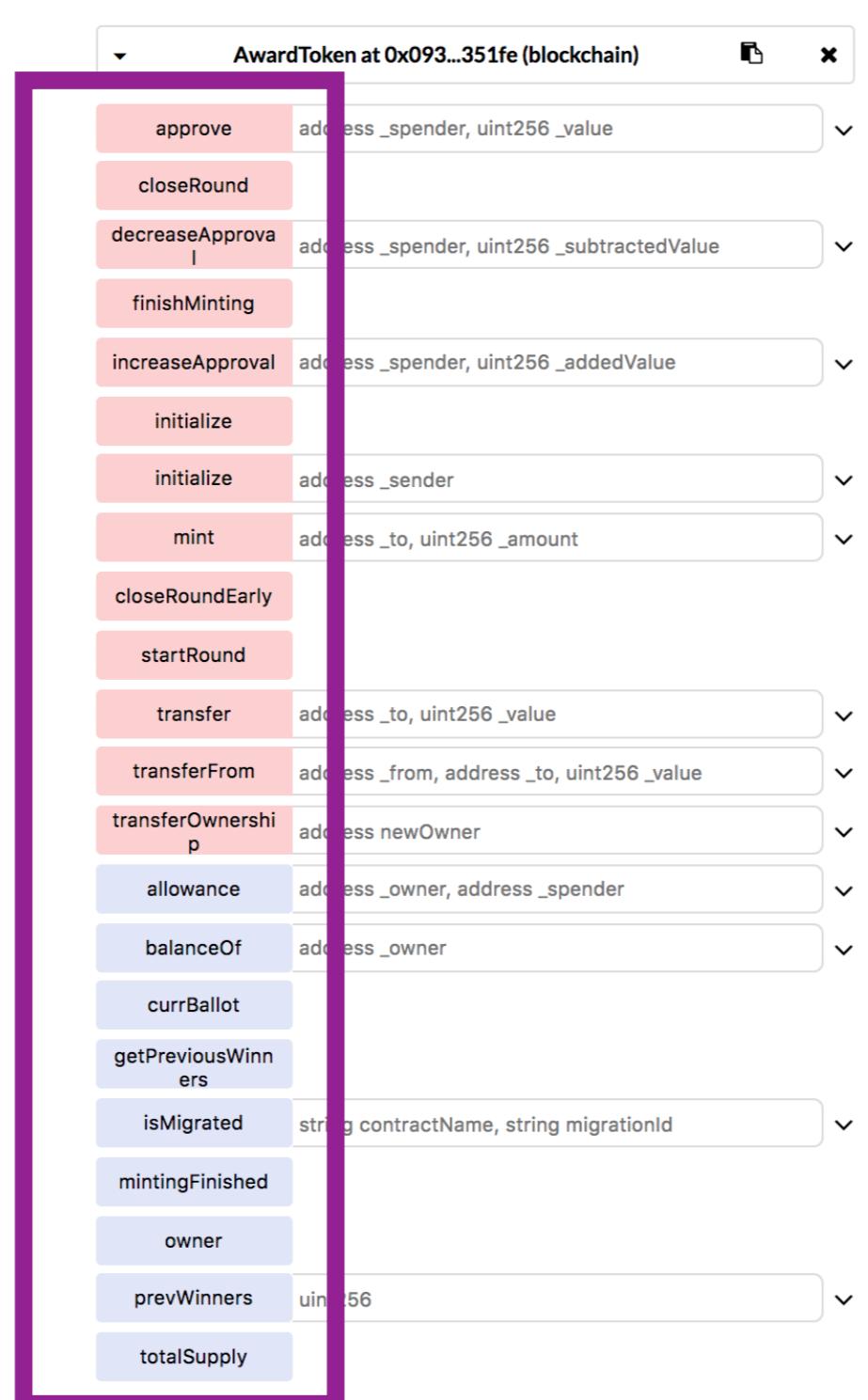


Voilà!

The Interactive UI for AwardToken.sol contract

[Update image](#)

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

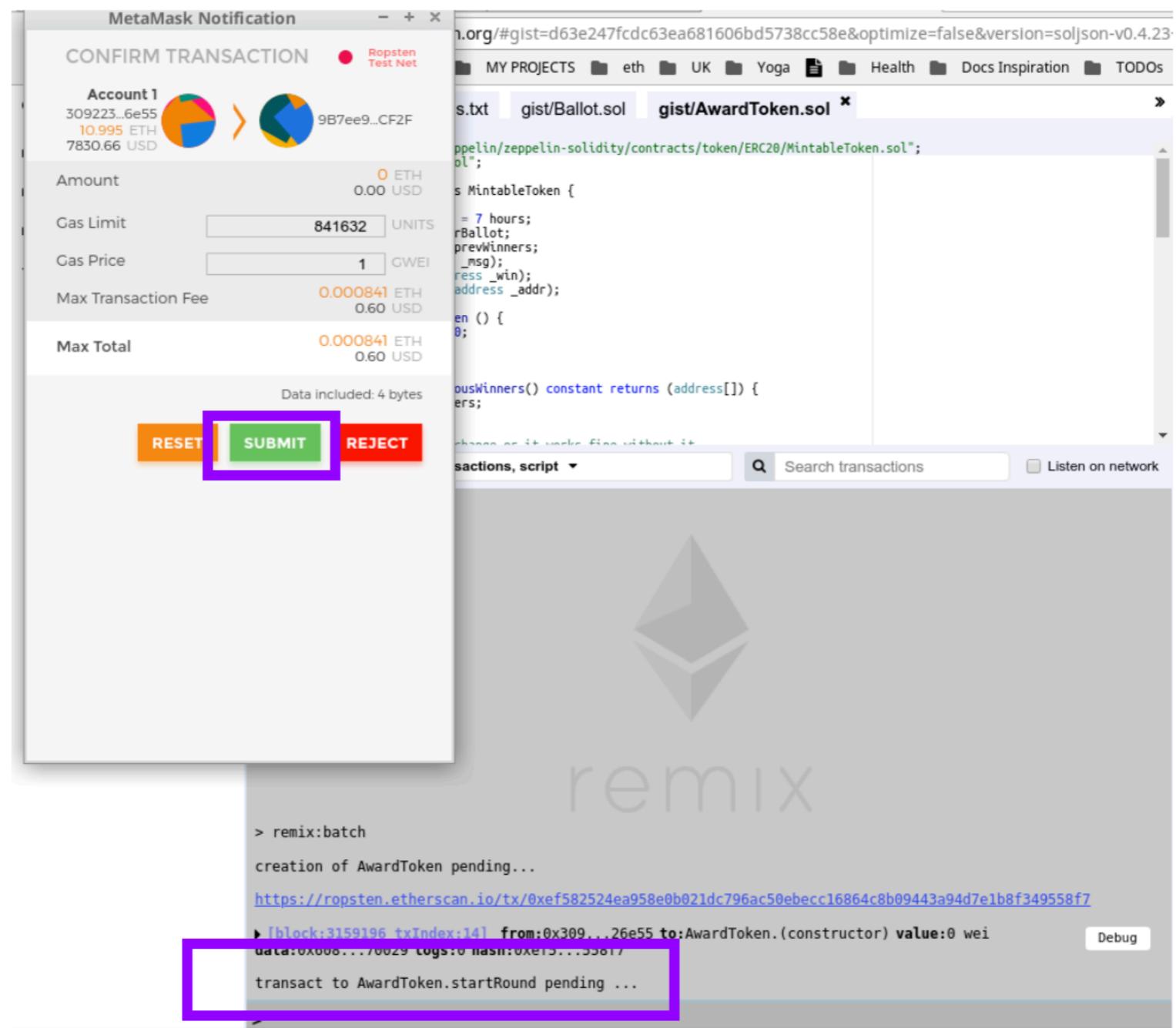


Execute startRound

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

```
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

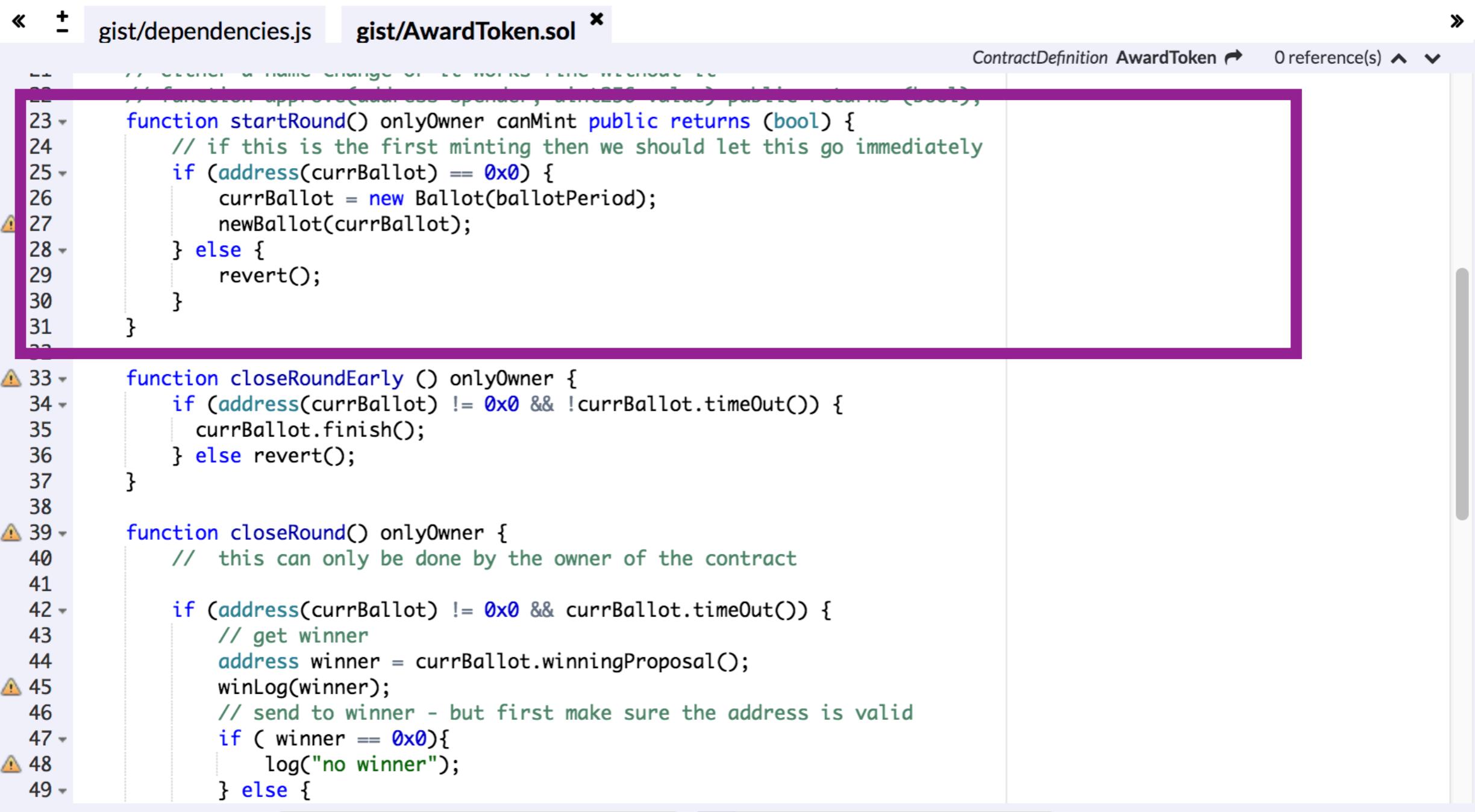
to see the logs

[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": "0x65f35fb257c91daed794331bfd2ad0f4439d49319d52a5b3bfb04c8496 9fdbeb", "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)
  ...
22 // Function approve(address spender, uint256 value) public returns (bool);
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 }
32
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

The screenshot shows a blockchain interface for the contract 'AwardToken at 0x9b7...0cf2f (blockchain)'. The interface lists various functions and their parameters. The 'currBallot' function is highlighted with a purple rectangular box. Other visible functions include approve, closeRound, decreaseApproval, finishMinting, increaseApproval, mint, startRound, transfer, transferFrom, transferOwnership, allowance, balanceOf, getPreviousWinners, mintingFinished, owner, prevWinners, and totalSupply.

Function	Description
approve	address _spender, uint256 _value
closeRound	
decreaseApproval	address _spender, uint256 _subtractedValue
finishMinting	
increaseApproval	address _spender, uint256 _addedValue
mint	address _to, uint256 _amount
startRound	
transfer	address _to, uint256 _value
transferFrom	address _from, address _to, uint256 _value
transferOwnership	address newOwner
allowance	address _owner, address _spender
balanceOf	address _owner
currBallot	
getPreviousWinners	
mintingFinished	
owner	
prevWinners	uint256
totalSupply	

Copy ballot's address

currBallot output

A screenshot of a blockchain interface showing the currBallot output of the AwardToken contract. The interface displays various functions and their parameters. The currBallot function is highlighted with a purple box, showing its output as the address: 0xE7bF60cee009DCDb2Ad8D045c19e76597bbF3c6.

Function	Description
approve	address _spender, uint256 _value
closeRound	
decreaseApproval	address _spender, uint256 _subtractedValue
finishMinting	
increaseApproval	address _spender, uint256 _addedValue
mint	address _to, uint256 _amount
startRound	
transfer	address _to, uint256 _value
transferFrom	address _from, address _to, uint256 _value
transferOwnership	address newOwner
allowance	address _owner, address _spender
balanceOf	address _owner
currBallot	: address: 0xE7bF60cee009DCDb2Ad8D045c19e76597bbF3c6
getPreviousWinner	s
mintingFinished	
owner	
prevWinners	uint256
totalSupply	

Switch to Ballot

(which is loaded from AwardToken.sol)

Run tab: dropdown

Compile Run Settings Analysis Debugger Support

Environment Injected Web3 🏃 Ropsten (3) ▾ i

Account 0x309...26e55 (10.994338592 ether) ▾ ✖ ✚

Gas limit 3000000

Value 0 wei ▾

Ballot ▾

Access Ballot contract

Paste address + click At Address

The screenshot shows a user interface for interacting with a blockchain application. At the top, there is a navigation bar with tabs: Compile, Run, Settings, Analysis, Debugger, and Support. The 'Run' tab is currently selected.

Below the navigation bar, there are several configuration fields:

- Environment:** Set to "Injected Web3" with a dropdown showing "Ropsten (3)" and an info icon.
- Account:** Set to "0x309...26e55 (10.994338592 ether)" with a dropdown, a copy icon, and a plus icon.
- Gas limit:** Set to "3000000".
- Value:** Set to "0" with a dropdown showing "wei".

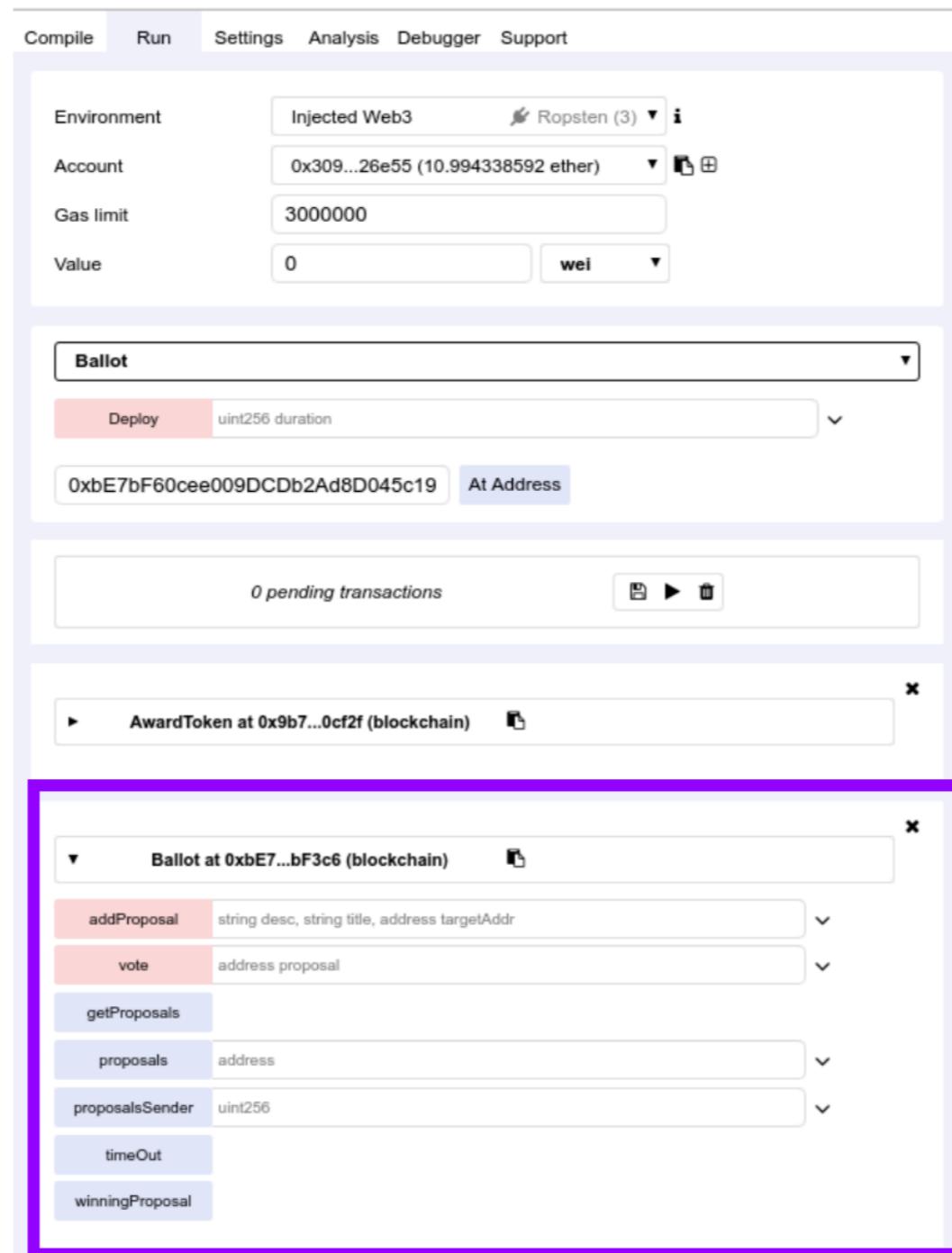
Below these settings, there is a section titled "Ballot" with a dropdown menu. Under "Ballot", there is a "Deploy" button and a field for "uint256 duration".

At the bottom of the interface, there are two buttons highlighted with purple boxes:

- A text input field containing the address "0xE7bF60cee009DCDb2Ad8D045c19".
- A button labeled "At Address".

See autogenerated UI

Interactive UI for Ballot.sol contract



Add a new proposal

Expand addProposal function

▼ **Ballot at 0xbE7...bF3c6 (blockchain)** 

×

addProposal	string desc, string title, address targetAddr	▼
vote	address proposal	▼
getProposals		
proposals	address	▼
proposalsSender	uint256	▼
timeOut		
winningProposal		

Copy your address

Run tab: Account

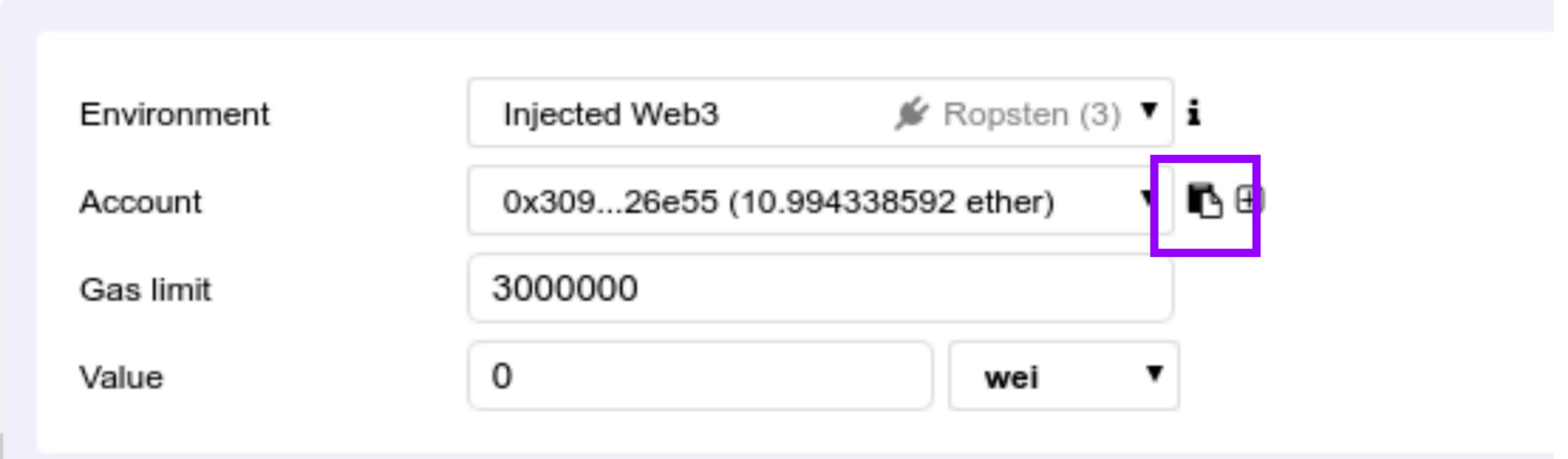
Compile Run Settings Analysis Debugger Support

Environment Injected Web3 Ropsten (3) ▾ i

Account 0x309...26e55 (10.994338592 ether) ▾ 

Gas limit 3000000

Value 0 wei ▾



Type a proposal

Run tab: Account

The screenshot shows the Remix IDE interface. At the top, it says "Ballot at 0xbE7...bF3c6 (blockchain)". Below that, there's a section titled "addProposal" which is highlighted with a purple border. Inside this section, there are two input fields: "desc:" containing the text "I think you could add a new feature to Remix that does..." and "title:" containing the text "This is my Remix improvements proposal". Below this, there's another input field for "targetAddr:" with the placeholder "address". At the bottom right, there's a pink button labeled "transact".

▼ Ballot at 0xbE7...bF3c6 (blockchain) ✖

addProposal ^

desc: "I think you could add a new feature to Remix that does..."

title: "This is my Remix improvements proposal"

targetAddr: address

transact

Add your address

Paste the address

▼ Ballot at 0xbE7...bF3c6 (blockchain)  X

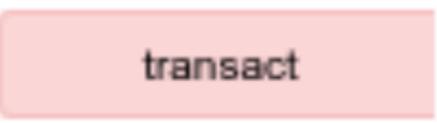
addProposal ^

desc: "I think you could add a new feature to Remix that does..."

title: "This is my Remix improvements proposal"

targetAddr: "0x3092232fb25e6b359a9fead9ed07ad752ff26e55"



 transact

Execute addProposal

transact button

▼ Ballot at 0xbE7...bF3c6 (blockchain)  X

addProposal ^

desc: "I think you could add a new feature to Remix that does..."

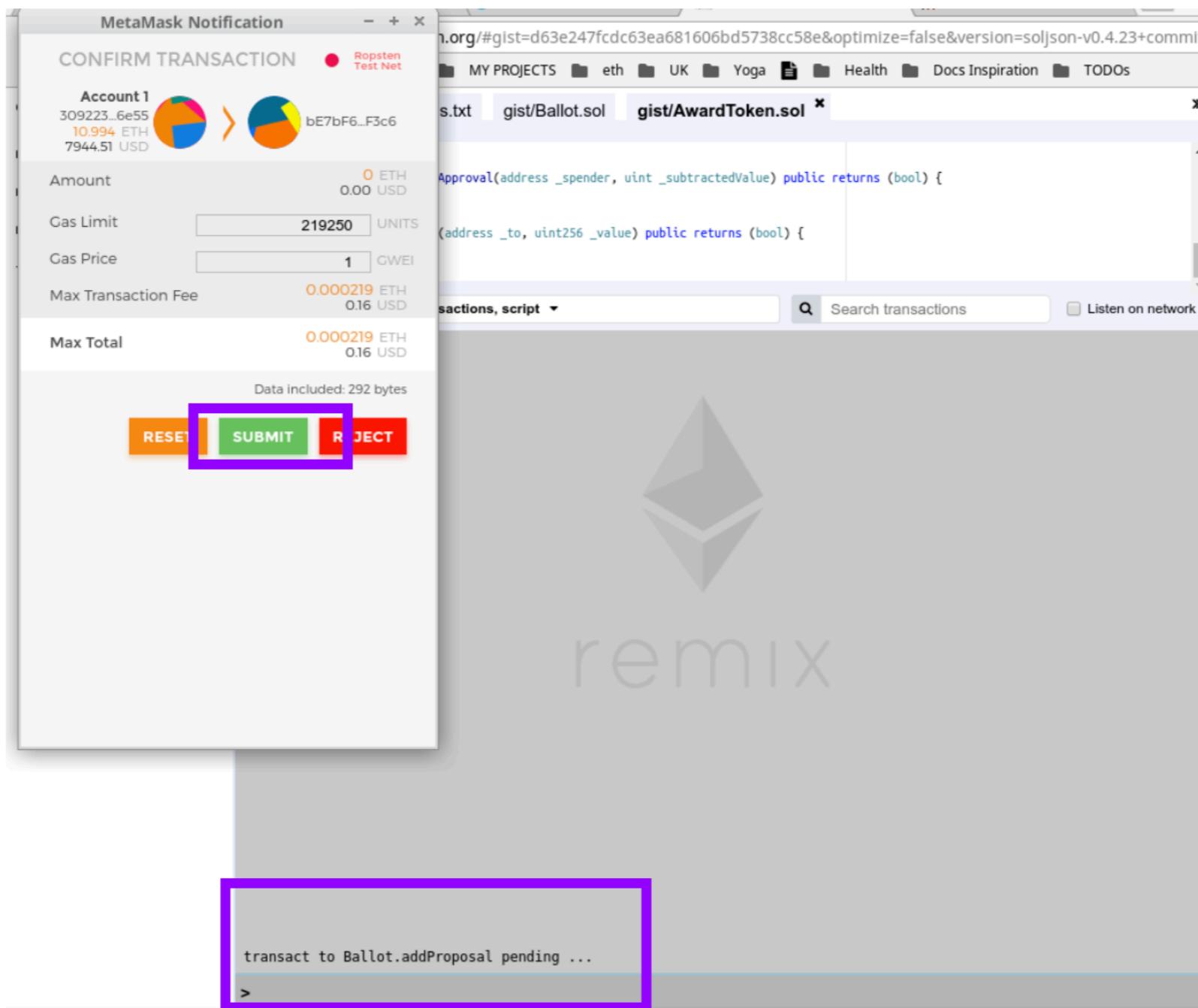
title: "This is my Remix improvements proposal"

targetAddr: "0x3092232fb25e6b359a9fead9ed07ad752ff26e55"



Confirm the transaction

Submit button



Check if tx succeeded

Terminal logs in Remix

Execute getProposals

getProposals call

The screenshot shows a blockchain ballot interface with the title "Ballot at 0xbE7...bF3c6 (blockchain)". Below the title, there is a list of methods:

- addProposal: string desc, string title, address targetAddr
- vote: address proposal
- getProposals: address proposals
- proposalsSender: uint256
- timeOut
- winningProposal

The "getProposals" method is highlighted with a purple rectangular border.

try it live!

See Proposals Addresses

well in so far there will only be 1 address

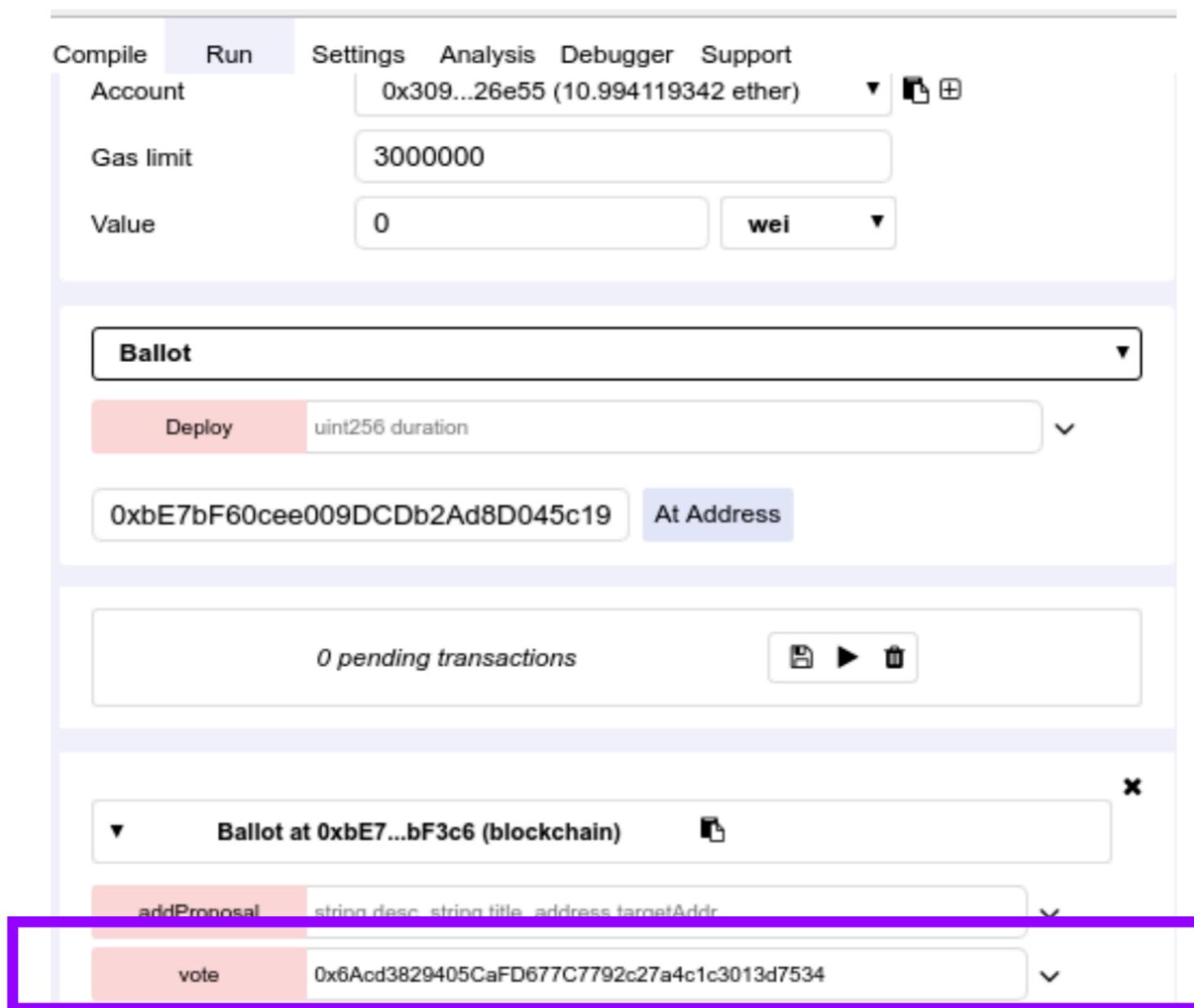
call to Ballot.getProposals

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

from	0x3092232fb25e6b359a9fead9ed07ad752ff26e55
to	Ballot.getProposals() 0xbE7bF60cee009DCDb2Ad8D045c19e76597bbF3c6
input	0x62564c48
decoded input	{}
decoded output	{ "0": "address[]: 0x3092232FB25e6b359a9fEad9eD07Ad752Ff26e55,0xFd0f51afb6 85Cd8735AfE7685D21355589602b8c,0x6Acd3829405CaFD677C7792c27a4c1c3013d7534" }
logs	[]

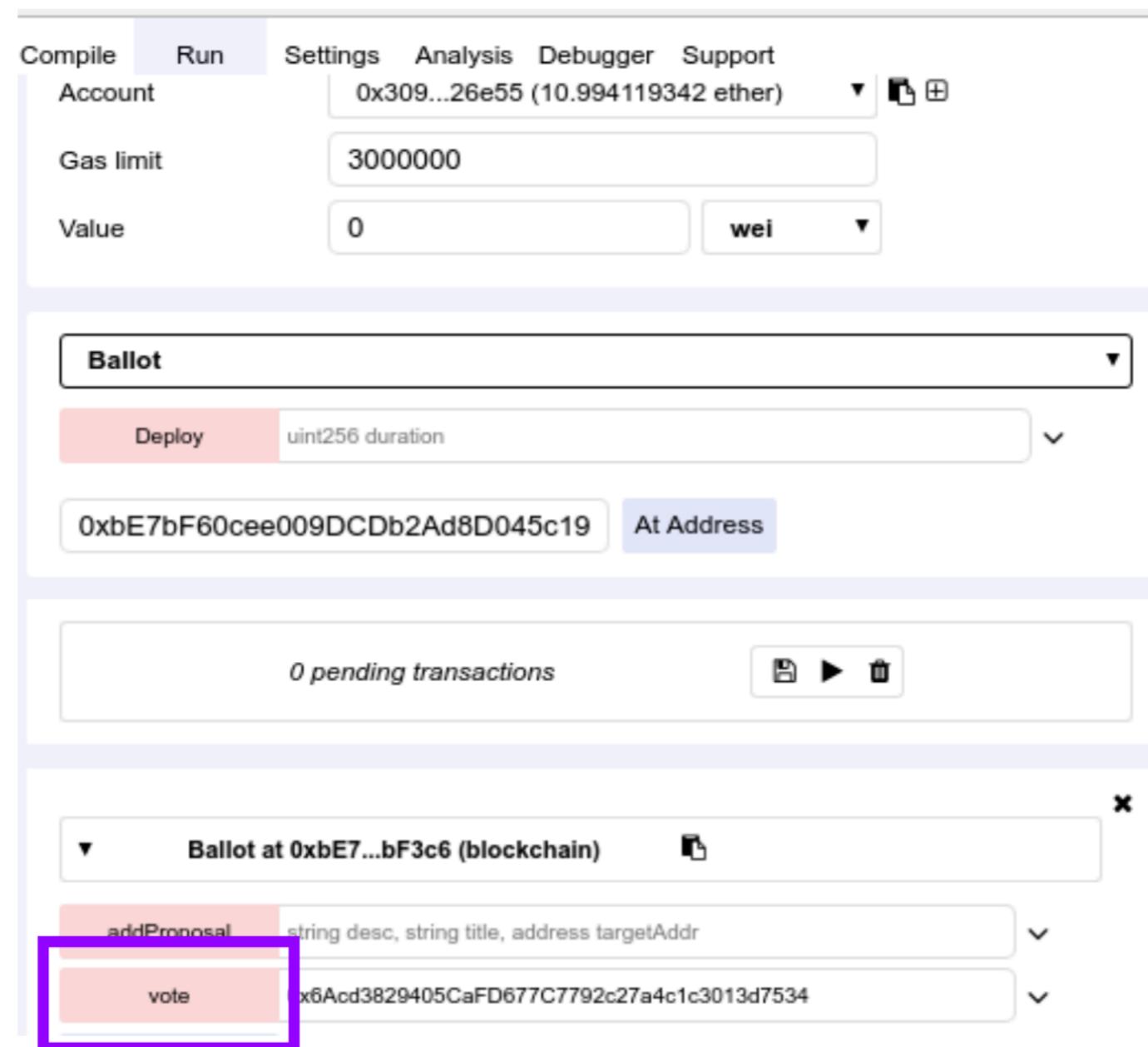
Vote for one Proposal

Paste Proposal Address you want to vote for



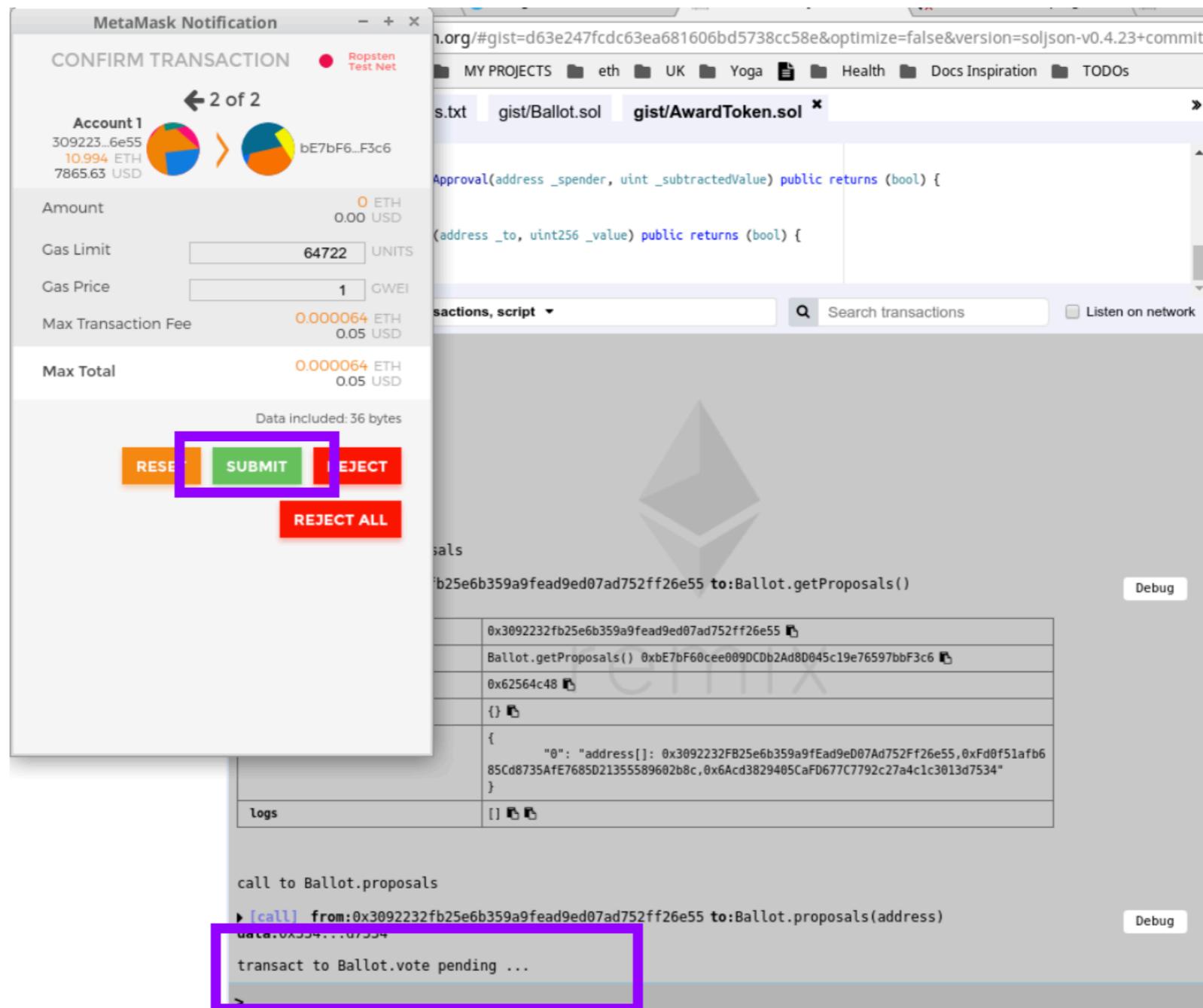
Execute vote transaction

vote button



Confirm the transaction

Submit button



Check if tx succeeded

Terminal logs in Remix

The screenshot shows the Ethereum Remix IDE interface. The top navigation bar has tabs for 'gist/dependencies.txt', 'gist/Ballot.sol', and 'gist/AwardToken.sol'. The sidebar on the left shows project files: 'browser', 'config', 'github', and 'gist' (containing 'AwardToken.sol', 'Ballot.sol', 'TUTORIAL.md', and 'dependencies.txt'). The main area displays terminal logs for a transaction.

Logs:

- call to Ballot.getProposals
 - ▶ [call] from:0x3092232fb25e6b359a9fead9ed07ad752ff26e55 to:Ballot.getProposals()
data:0x625...64c48
 - from: 0x3092232fb25e6b359a9fead9ed07ad752ff26e55
 - to: Ballot.getProposals() 0xbE7bF60cee009DCDb2Ad8D045c19e76597bbF3c6
 - input: 0x62564c48
 - decoded input: {}
 - decoded output:

```
{ "0": "address[]: 0x309223FB25e6b359a9fEad9eD07Ad752Ff26e55,0xFd0f51afb685Cd8735AfE7685D21355589602b8c,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

Now let's try it out connecting a frontend

<http://bit.ly/remix-voting>

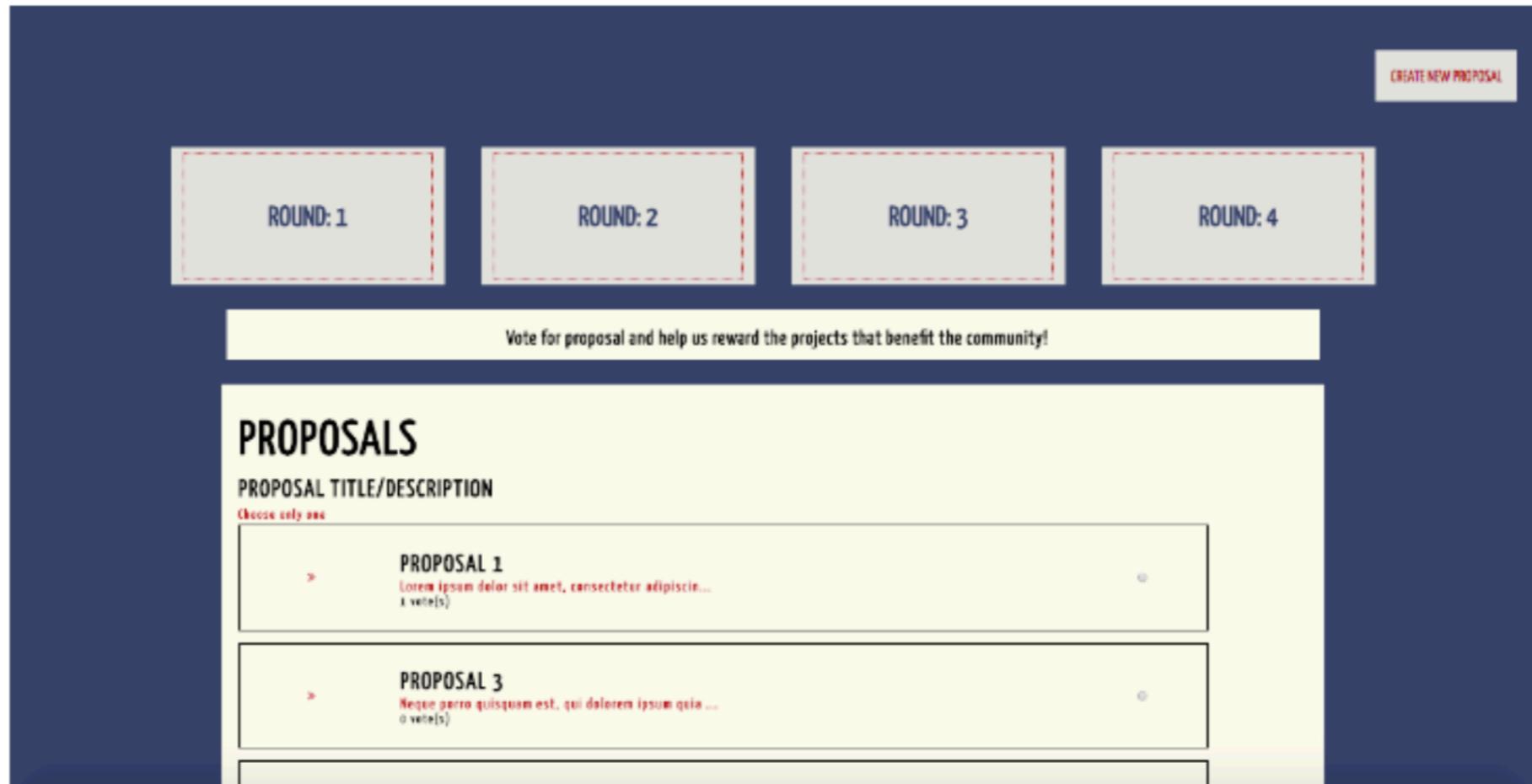
To access our Award Token from this frontend -
you need the address of the Award Token.

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;  
    }  
    Proposal[] proposals;  
}
```



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



CREATE NEW PROPOSAL

ROUND:1 ROUND:2 ROUND:3 ROUND:4

Vote for proposal and help us reward the projects that benefit the community!

PROPOSALS

PROPOSAL TITLE/DESCRIPTION
Choose only one

PROPOSAL 1
Lorem ipsum dolor sit amet, consectetur adipiscing...
1 vote(s)

PROPOSAL 2
Neque porro quisquam est, qui dolorem ipsum quia...
0 vote(s)

PROPOSAL 3
Neque porro quisquam est, qui dolorem ipsum quia...
0 vote(s)

Let's check results

<http://bit.ly/remix-voting>

Check the state of the contract

Ballot at 0x712...0Aa64 (blockchain)

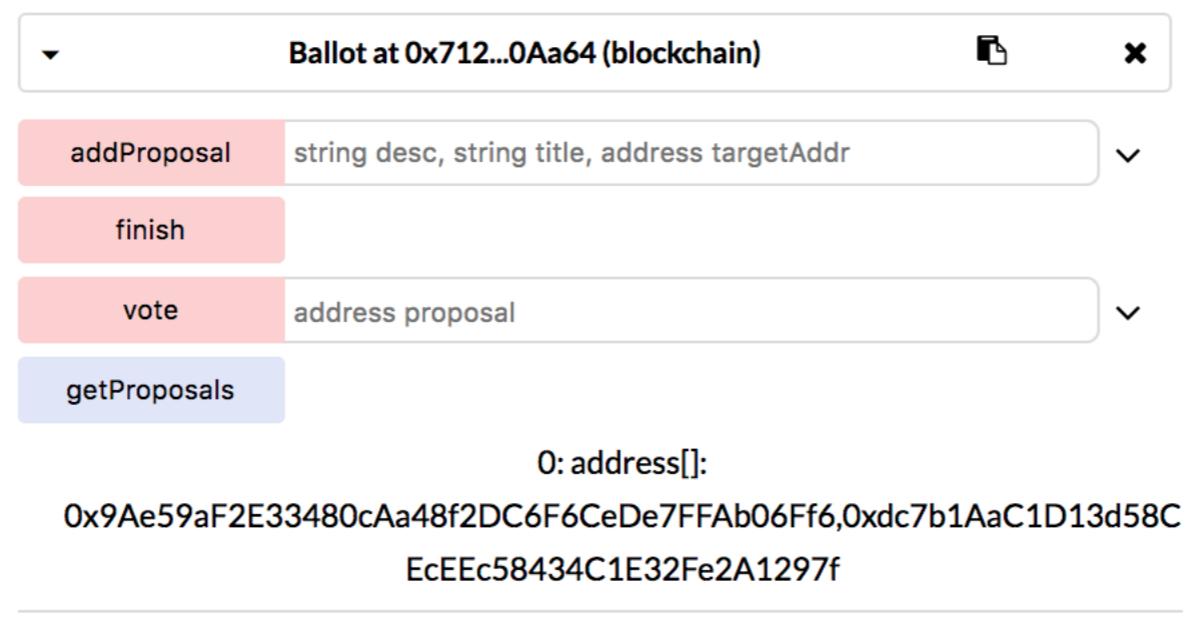
addProposal string desc, string title, address targetAddr

finish

vote address proposal

getProposals

O: address[]:
0x9Ae59aF2E33480cAa48f2DC6F6CeDe7FFAb06Ff6,0xdc7b1AaC1D13d58C
EcEEc58434C1E32Fe2A1297f



@ninabreznik @ryestew @yann300 @serapath @iurimatias

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