

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>



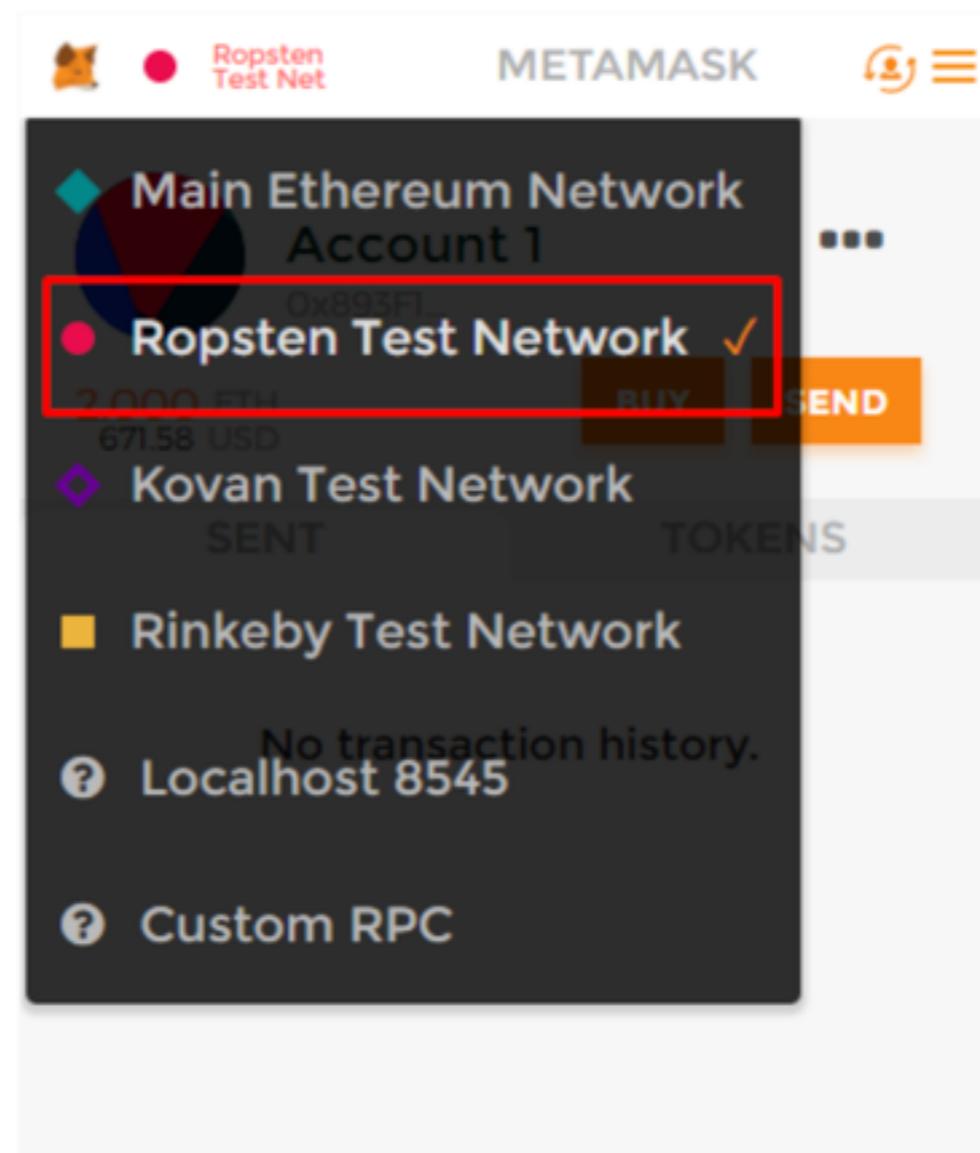
Install Metamask

chrome.google.com/webstore

The image shows a screenshot of the Chrome Web Store page for the MetaMask extension. The page is titled "MetaMask" and is offered by <https://metamask.io>. It has a 5-star rating from 983 reviews and is categorized as "Productivity" with 1,082,276 users. The page includes a "FEATURES" section with options like "Runs Offline", "By Google", "Free", "Available for", and "Works with". There are also "RATINGS" and "RELATED" sections. The "RELATED" section includes "MyEtherWallet" (5 stars, 260 reviews), "Disable Extensions Temporarily" (5 stars, 250 reviews), "EtherAddressLookup" (5 stars, 65 reviews), and "The FFZ Add-On Pack" (5 stars, 117 reviews). The main content area shows a preview of the extension's interface, which includes a "Main" section with a balance of 0.867140 ETH and a "HISTORY" section with transactions from November 2016. The page also includes a "Description" section that explains that MetaMask is an Ethereum Browser Extension that injects the Ethereum web3 API into every website's javascript context, allowing users to access Ethereum enabled distributed applications, or "Dapps", in their normal Chrome browser. The page also includes a "Compatible with your device" section and a "Additional Information" section with details like Version: 4.6.1, Updated: April 30, 2018, and Size: 5.58MiB.

Login to Metamask

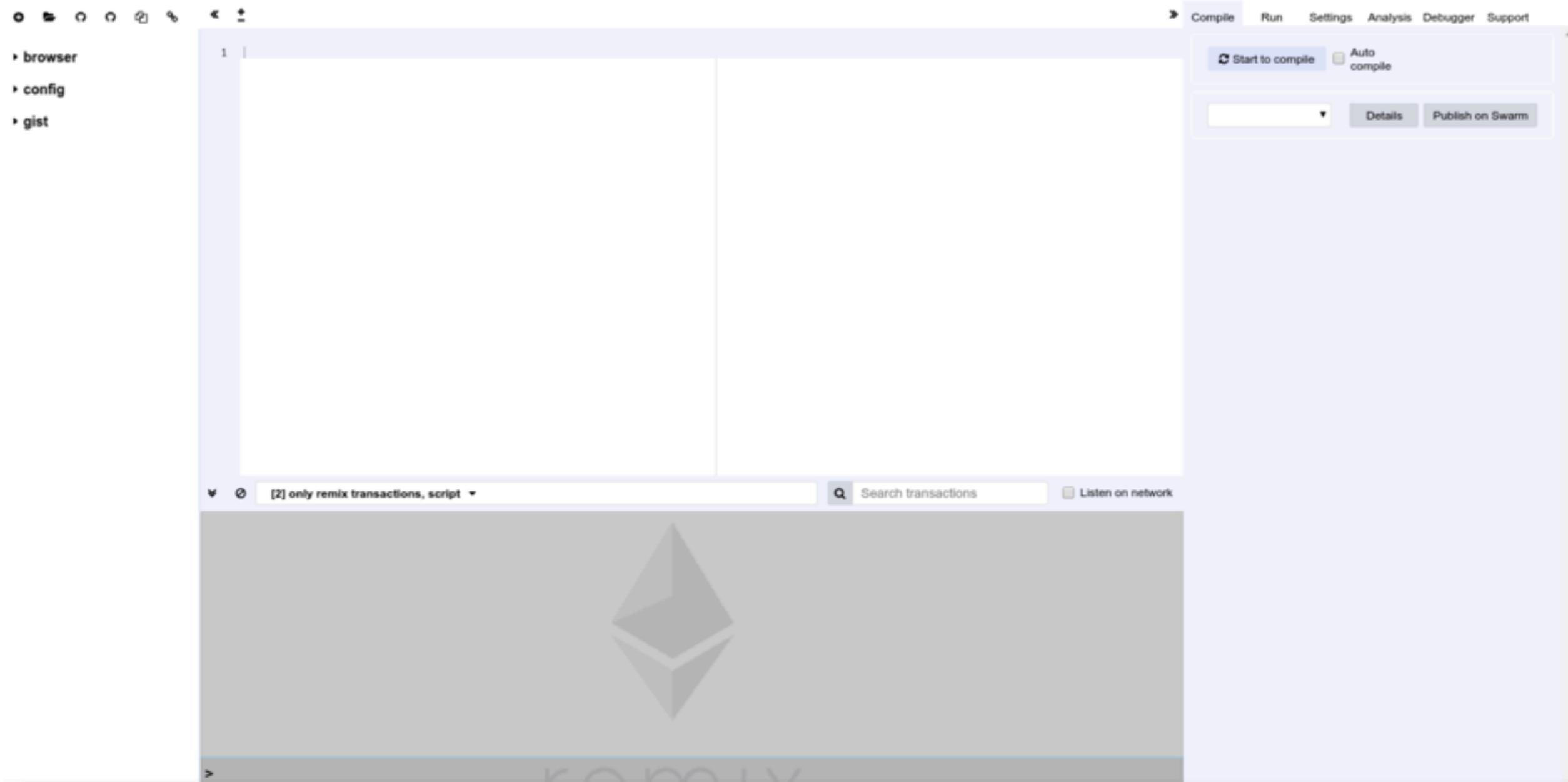
Ropsten Test Network



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)



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

```
1 pragma solidity ^0.4.0;
2 contract Ballot {
3
4     struct Voter {
5         uint weight;
6         bool voted;
7         uint8 vote;
8         address delegate;
9     }
10    struct Proposal {
11        uint voteCount;
12    }
13
14    address chairperson;
15    mapping(address => Voter) voters;
16    Proposal[] proposals;
17
```

Editor



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 a function ballot(uint8 _numProposals) public ... (Relevant source part starts here and spans ...)

```
- 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 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

>
```

Terminal

Console

Run Tab

Compile **Run** Settings Analysis Debugger Support Test

Environment ✂ Ropsten (3) ⓘ

Account 📄 ⊕

Gas limit

Value 🗨️ 1 ⬆

⬆

Transactions recorded: 🔔 4 ⌵

Deployed Contracts 🗑️

⌵ 📄 ✕

<input type="button" value="approve"/>	<input type="text" value="address _spender, uint256 _value"/>	⌵
<input type="button" value="closeRound"/>		
<input type="button" value="closeRoundEarly"/>		
<input type="button" value="decreaseApproval"/>	<input type="text" value="address _spender, uint256 _subtractedValue"/>	⌵
<input type="button" value="finishMinting"/>		
<input type="button" value="increaseApproval"/>	<input type="text" value="address _spender, uint256 _addedValue"/>	⌵
<input type="button" value="mint"/>	<input type="text" value="address _to, uint256 _amount"/>	⌵

Universal DAPP
UI to the Contract

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

```
1 pragma solidity ^0.4.0;
2 contract Ballot {
3
4     struct Voter {
5         uint weight;
6         bool voted;
7         uint8 vote;
8         address delegate;
9     }
10    struct Proposal {
11        uint voteCount;
12    }
13
14    address chairperson;
15    mapping(address => Voter) voters;
16    Proposal[] proposals;
17
```

Compile Run Settings Analysis Debugger Support Test

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 a 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 environment 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 = Injected web3
(Ropsten)

The screenshot shows a Solidity IDE interface. On the left, a code editor displays the source code for a contract named 'AwardToken'. The code includes imports for 'MintableToken' and 'Ballot', and defines several functions: 'AwardToken' (constructor), 'getPreviousWinners', and 'startRound'. On the right, the 'Run' tab is active, showing deployment settings. The 'Environment' dropdown is set to 'Injected Web3' and is highlighted with a purple box. Below it, the 'Account' is set to '0x667...d091d (1.453587112999635446)', 'Gas limit' is '3000000', and 'Value' is '0'. A dropdown menu shows the contract name 'AwardToken'. A red 'Deploy' button is visible, along with a 'Load contract from Address' button and an 'At Address' button.

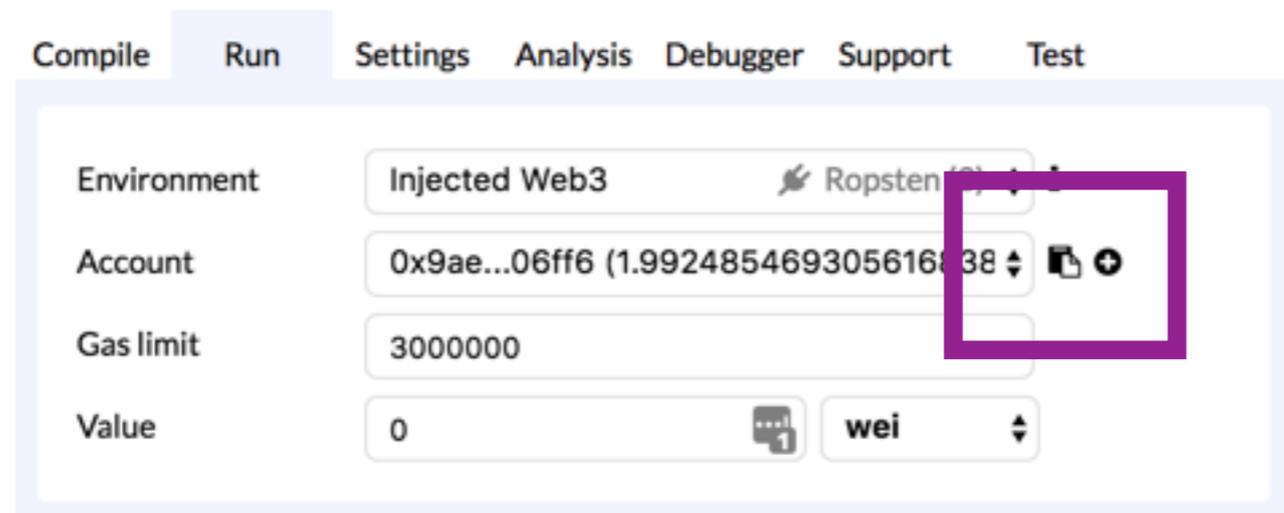
```
1 import "github/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/MintableToken.sol";
2 import "gist/Ballot.sol";
3
4 contract AwardToken is MintableToken {
5     uint quantity;
6     uint ballotPeriod = 7 hours;
7     Ballot public currBallot;
8     address[] public prevWinners;
9     event 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 on immediately
25    }
26
```

Get some TEST ether

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

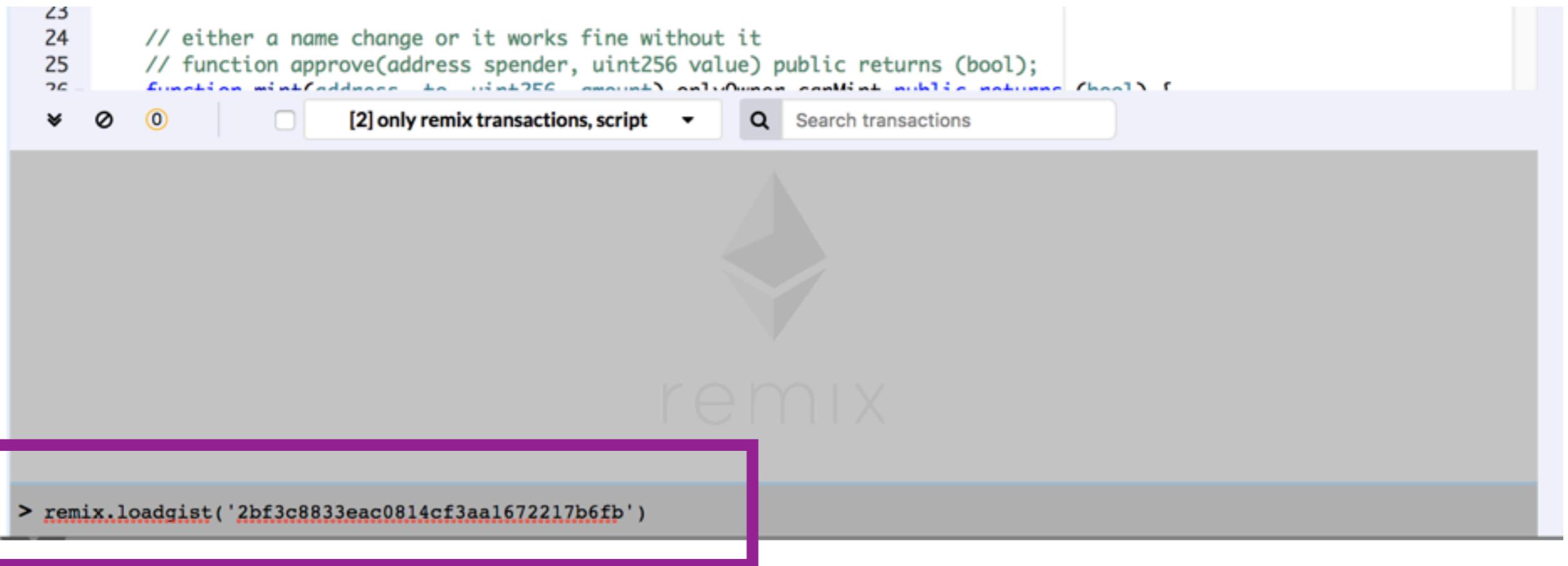
But FIRST:

Copy your address - here or in Metamask



Load files to Remix

```
remix.loadgist('1b87ded5087790b67b5c4cd90a68065f')
```



here in the console

Open file

gist/dependencies.js



- ▶ browser
- ▶ config
- ▾ gist
 - AwardToken.sol
 - Ballot.sol
 - README.md
 - dependencies.js

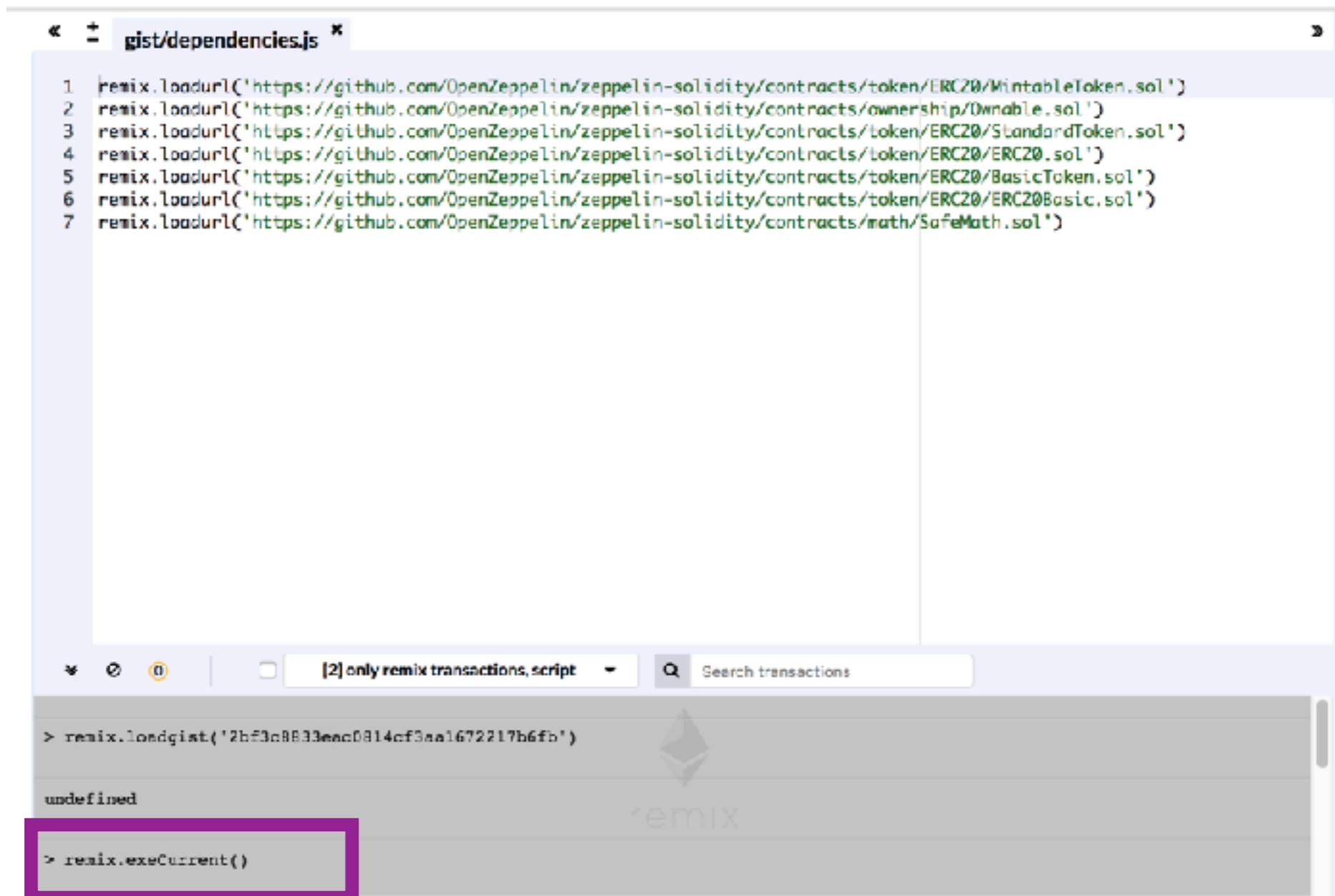
« + gist/dependencies.js x

```
1 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/MintableToken.sol')
2 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/ownership/Ownable.sol')
3 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/StandardToken.sol')
4 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/ERC20.sol')
5 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/BasicToken.sol')
6 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/ERC20Basic.sol')
7 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/math/SafeMath.sol')
```

Load dependencies

`remix.exeCurrent()`

(when dependencies.js is the active file)



```
gist/dependencies.js
1 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/MintableToken.sol')
2 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/ownership/Ownable.sol')
3 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/StandardToken.sol')
4 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/ERC20.sol')
5 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/BasicToken.sol')
6 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/ERC20Basic.sol')
7 remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/math/SafeMath.sol')

[2] only remix transactions, script
Search transactions

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

See new folder

github/OpenZeppelin/openzeppelin-zos/contracts

The image shows a code editor interface with a file explorer on the left and a code editor on the right. The file explorer shows a tree structure with the following folders and files:

- browser
- config
- github (highlighted in a purple box)
 - OpenZeppelin
 - openzeppelin-zos
 - contracts
 - token
 - ownership
 - math
- gist
 - AwardToken.sol
 - Ballot.sol
 - README.md
 - dependencies.js

The code editor shows the following code:

```
1 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-zos/contracts/token/ERC20/MintableToken.sol')
2 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-zos/contracts/ownership/Ownable.sol')
3 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-zos/contracts/token/ERC20/StandardToken.sol')
4 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-zos/contracts/token/ERC20/ERC20.sol')
5 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-zos/contracts/token/ERC20/BasicToken.sol')
6 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-zos/contracts/token/ERC20/ERC20Basic.sol')
7 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-zos/contracts/math/SafeMath.sol')
```

Try to compile AwardToken.sol

- Click AwardToken in the File Explorer to make it the active file in the Editor
- Compile tab: Start to compile button

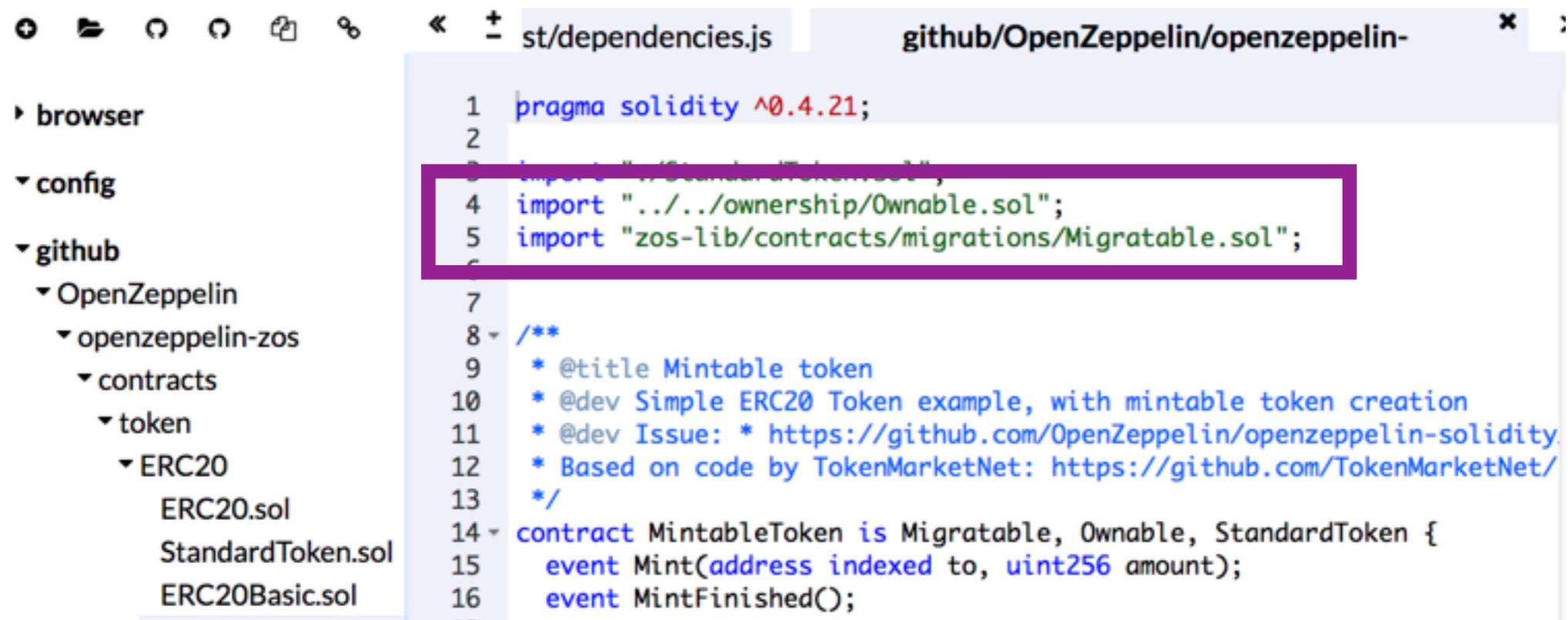
But it won't work

The screenshot shows a Solidity IDE interface. On the left, a file explorer shows a project structure with a 'gist' folder containing 'AwardToken.sol'. The main editor displays the code for 'AwardToken.sol', which includes imports for 'ERC20' and 'Ballot', and defines a contract 'AwardToken' that inherits from 'MintableToken'. The code includes variables for 'quantity' and 'ballotPeriod', and functions for 'AwardToken()', 'getPreviousWinners()', and 'startRound()'. On the right, the 'Compile' tab is active, showing the current compiler version as '0.4.25+commit.59dbf8f1.Emscripten.clang'. Below this, there are options for 'Select new compiler version', 'Auto compile' (checked), 'Enable Optimization', and 'Hide warnings'. A 'Start to compile' button is visible. At the bottom of the compile panel, a red error message states: 'file provider localhost not available while trying to...'. The 'Compile' button and the error message are highlighted with purple boxes.

```
1 import "github/OpenZeppelin/openzeppelin-zos/contracts/token/ERC20/ERC20.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 on i
```

AwardToken is importing MintableToken.sol which is trying to import a file inside of zos-lib.

We need to get the zos-lib from NPM and make the NPM folder accessible from Remix



```
1 pragma solidity ^0.4.21;
2
3 import "StandardToken.sol";
4 import " ../../ownership/Ownable.sol";
5 import "zos-lib/contracts/migrations/Migratable.sol";
6
7
8 /**
9  * @title Mintable token
10 * @dev Simple ERC20 Token example, with mintable token creation
11 * @dev Issue: * https://github.com/OpenZeppelin/openzeppelin-solidity
12 * Based on code by TokenMarketNet: https://github.com/TokenMarketNet/
13 */
14 contract MintableToken is Migratable, Ownable, StandardToken {
15     event Mint(address indexed to, uint256 amount);
16     event MintFinished();
```

Setup to access local files 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)

```
npm install remixd  
npm install zos-lib
```

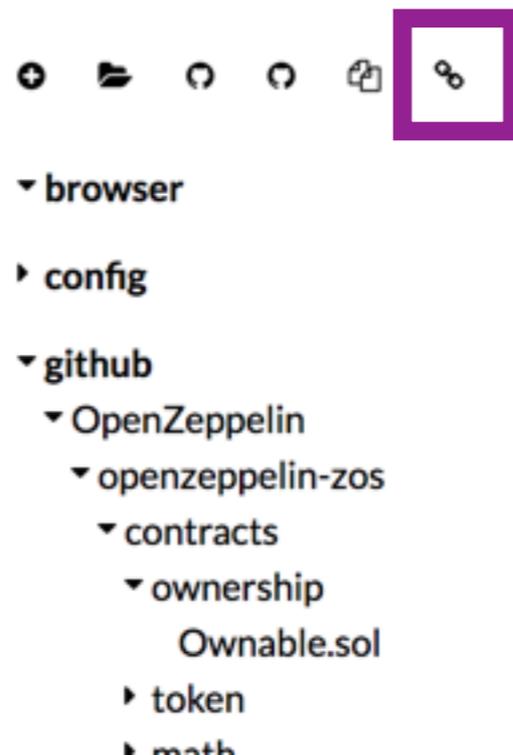
Get Remixd Going!

In a Terminal / Console
from your project's directory

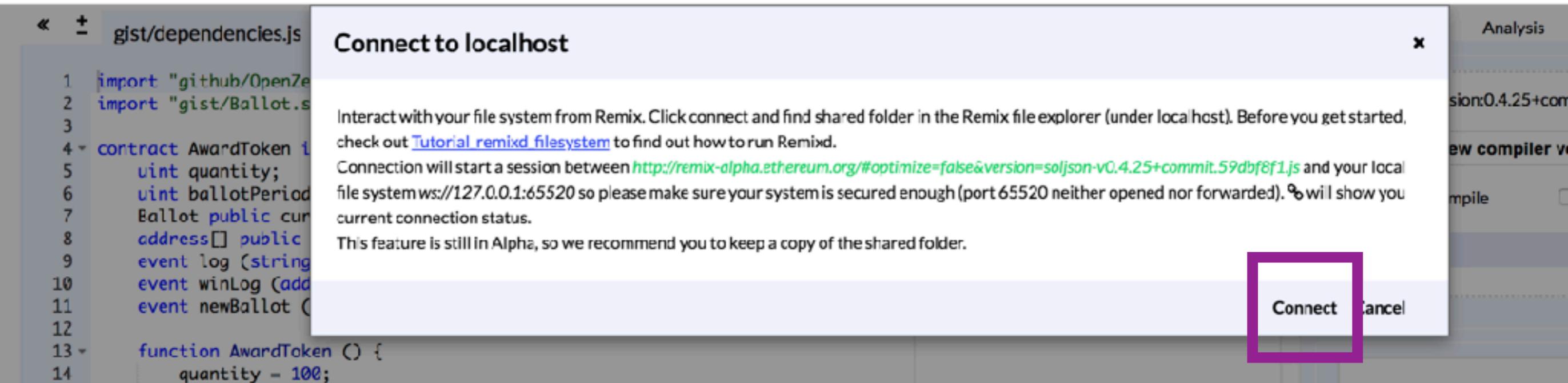
type:

```
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)



Click 'Connect'



Then the link icon should turn green.

Compile AwardToken

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

Click the “Start to compile” button

```
1 import "github/OpenZeppelin/openzeppelin-zos/contracts/token/ERC20/MinimalERC20.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
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        }
29    }
30}
```

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

Select new compiler version

Auto compile Enable Optimization Hide warnings

Start to compile

AwardToken

Details ABI Bytecode

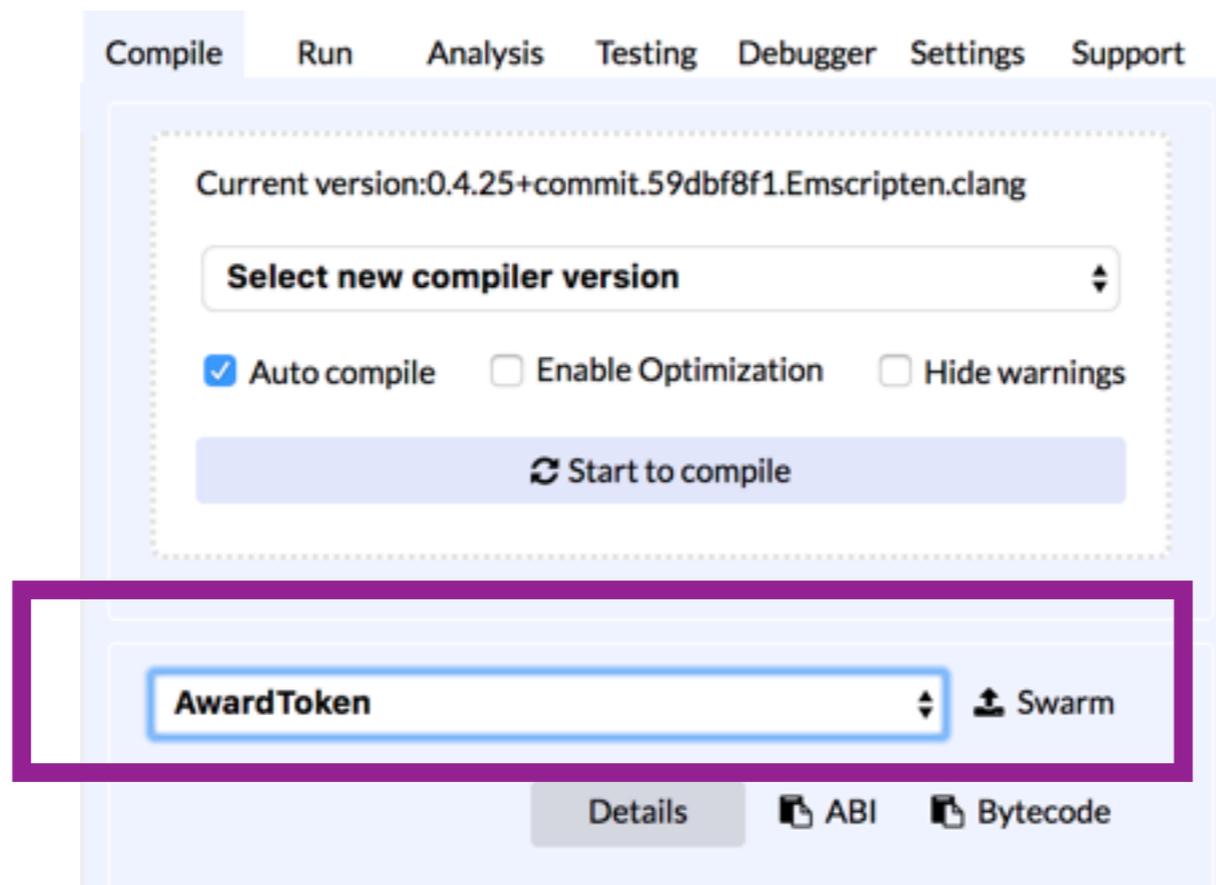
Static Analysis raised 52 warning(s) that requires your attention. Click here to show the warning(s).

gist/Ballot.sol:26:5: Warning: Defining constructors for a contract that already has a constructor. This is deprecated and will be removed in a future version of Solidity. (Relevant source part starts here and spans across multiple lines)

```
function Ballot(uint duration) public {
```

See compiled contracts

AwardToken + all it's imported contracts



Imported Contracts

The screenshot displays a web3 development interface with the following components:

- Navigation Bar:** Compile, Run (selected), Settings, Analysis, Debugger, Support, Test.
- Environment:** Injected Web3, Ropsten (3)
- Account:** 0x9ae...06ff6 (1.992485469305616838)
- Gas limit:** 3000000
- Value:** 0, wei

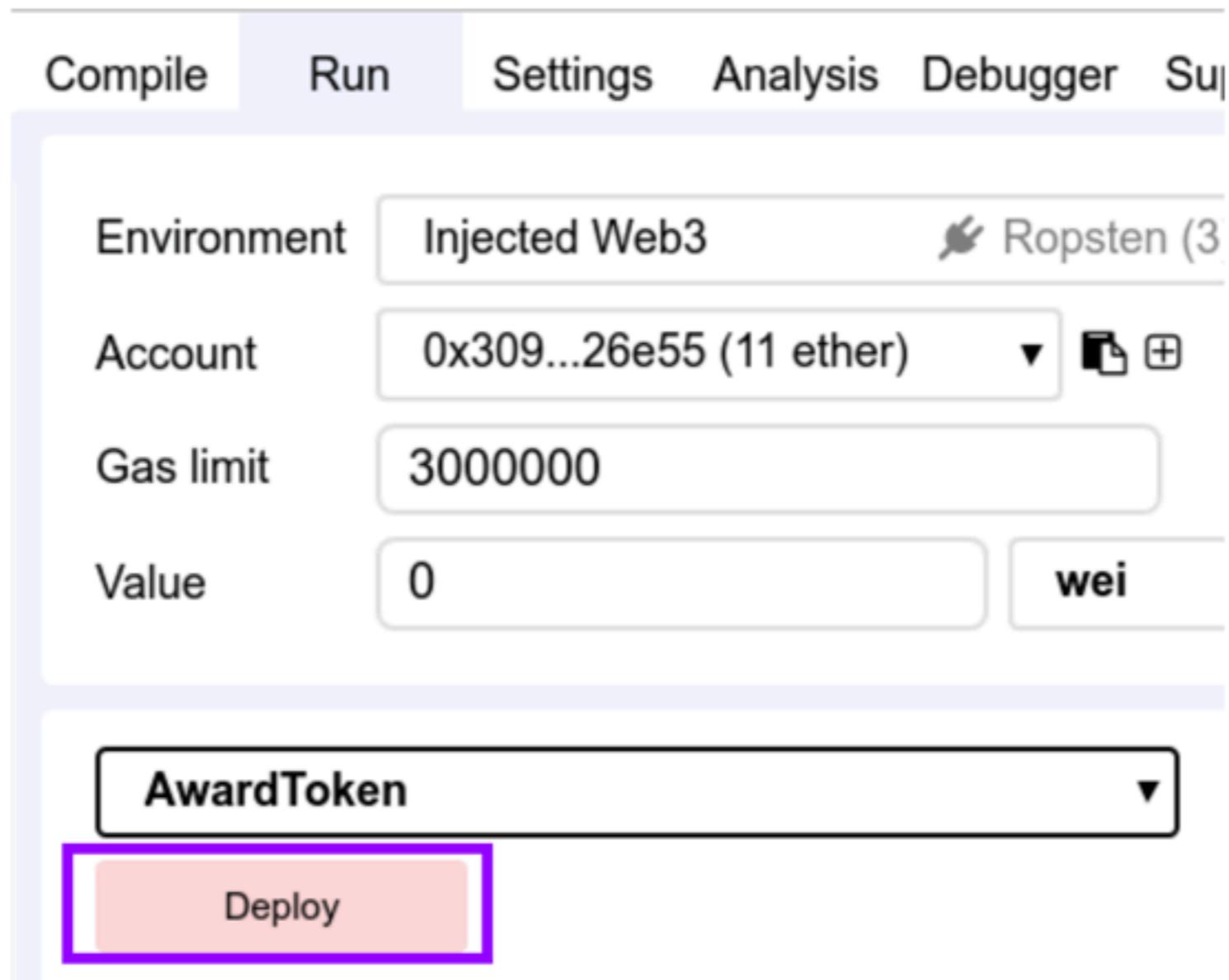
A dropdown menu is open, showing a list of imported contracts:

- ✓ AwardToken (selected)
- Ballot
- SafeMath
- Ownable
- BasicToken
- ERC20
- ERC20Basic
- MintableToken
- StandardToken

At the bottom, there is a section labeled "Deployed Contracts" with a trash icon.

Deploy the contract

Run tab: Deploy button



The screenshot shows the 'Run' tab of a development tool. At the top, there are tabs for 'Compile', 'Run', 'Settings', 'Analysis', 'Debugger', and 'Su'. Below these, the 'Environment' is set to 'Injected Web3' with a dropdown arrow and 'Ropsten (3' next to it. The 'Account' is '0x309...26e55 (11 ether)' with a dropdown arrow and a plus icon. The 'Gas limit' is '3000000'. The 'Value' is '0' with a unit dropdown set to 'wei'. At the bottom, there is a dropdown menu for the contract name, currently showing 'AwardToken'. Below the dropdown is a red 'Deploy' button, which is highlighted with a purple border.

Environment	Injected Web3	✎ Ropsten (3
Account	0x309...26e55 (11 ether)	▼ 📄 ⊕
Gas limit	3000000	
Value	0	wei

AwardToken ▼

Deploy

Confirm the transaction

Submit button

But make sure you put in a gas price!

The image shows a screenshot of a web browser displaying the Remix IDE interface. A MetaMask notification overlay is visible on the left side, titled "CONFIRM TRANSACTION". The notification shows the account details for "Account 1" (309223...6e55) with 11,000 ETH (7921.32 USD). The transaction details include an amount of 0 ETH (0.00 USD), a gas limit of 2409888 UNITS, a gas price of 1 GWEI, and a max transaction fee of 0.002409 ETH (1.73 USD). The notification also shows a "SUBMIT" button highlighted with a purple box.

The main interface shows the Remix IDE with the following components:

- Environment:** Injected Web3, Ropsten (3)
- Account:** 0x309...26e55 (11 ether)
- Gas limit:** 3000000
- Value:** 0 wei
- Contract:** AwardToken
- Buttons:** Deploy, Load contract from Address, At Address
- Transactions:** 1 pending transactions
- Contract Instances:** 0 contract instances

The Solidity code in the editor is as follows:

```
import "github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/MintableToken";
import "gist/Ballot.sol";

contract AwardToken is MintableToken {
    uint quantity;
    uint ballotPeriod = 7 hours;
    Ballot public currBallot;
    address[] public prevWinners;
    event log (string _msg);
    event winLog (address _win);
    event newBallot (address _addr);

    function AwardToken () {
        quantity = 100;
    }

    function getPreviousWinners() constant returns (address[]) {
        return prevWinners;
    }
}
```

The terminal at the bottom shows the following commands and output:

```
> remix:loadgist 1483e5599012c3783def91ead259ece8
> remix:batch
creation of AwardToken pending...
```

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

The screenshot displays a web development tool interface with the following components:

- Navigation:** Compile, Run, Settings, Analysis, Debugger, Support, Test
- Environment:** Injected Web3, Ropsten (3)
- Account:** 0x9ae...06ff6 (1.993121706305616838)
- Gas limit:** 3000000
- Value:** 0, wei
- Contract Selection:** AwardToken
- Buttons:** Deploy, Load contract from Address, At Address
- Transactions:** Transactions recorded: 1
- Deployed Contracts:** AwardToken at 0x574...40360 (blockchain)

A purple box highlights the right-pointing arrow icon next to the 'AwardToken at 0x574...40360 (blockchain)' entry in the 'Deployed Contracts' section.

Voilà!

The Interactive UI for AwardToken.sol contract

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



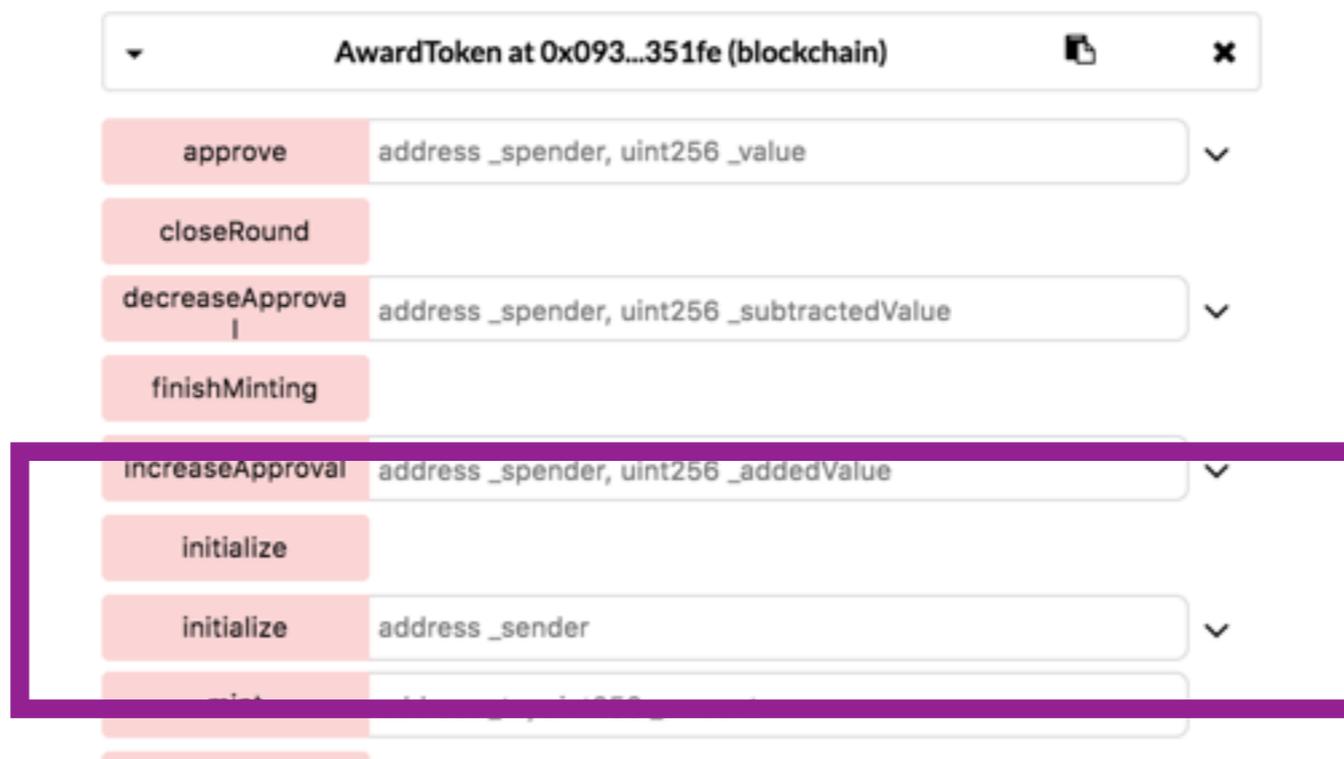
The screenshot displays the interactive UI for the AwardToken.sol contract, titled "AwardToken at 0x093...351fe (blockchain)". The interface lists various methods and their parameters, categorized into two groups: methods (pink buttons) and variables (blue buttons). A purple box highlights the methods section.

Method	Parameters
approve	address _spender, uint256 _value
closeRound	
decreaseApproval	address _spender, uint256 _subtractedValue
finishMinting	
increaseApproval	address _spender, uint256 _addedValue
initialize	
initialize	address _sender
mint	address _to, uint256 _amount
closeRoundEarly	
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	
isMigrated	string contractName, string migrationId
mintingFinished	
owner	
prevWinners	uint256
totalSupply	

Initialize

There are 2 initialize functions. Both from the imported Zeppelin contracts. And we need now need to run 1 of them but not the other.

Choose the one with that needs an address and copy your address and past it in there.



Run Initialize

In the expanded (see below) or collapsed Initialize method - fire it up and pay for the transaction.

The screenshot shows the 'Run' tab in a development tool. The configuration is as follows:

Field	Value
Environment	Injected Web3 (Ropsten)
Account	0x18f...bd131 (4.418569494631537502)
Gas limit	3000000
Value	0 wei

A purple box highlights the account selection area, which includes a dropdown menu and icons for copying and editing the account address.

The screenshot shows the 'initialize' method call in a debugger. The parameters are:

- `_sender:` 0x18f8a7184d5c6a1cb6a0eba926b99d7201bd131

A blue box highlights the sender address, and a purple box highlights the 'transact' button, which is used to execute the transaction.

Execute startRound

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

A screenshot of a blockchain explorer interface showing a list of functions for 'AwardToken at 0x9b7...0cf2f (blockchain)'. The 'startRound' function is highlighted with a purple box. The interface includes a search bar at the top, a list of functions with their parameters, and a 'startRound' function highlighted with a purple box. The functions listed are:

- 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

Confirm the transaction

The image shows a MetaMask notification window for a transaction confirmation on the Ropsten Test Net. The window displays account information for Account 1 (309223...6e55) and the recipient account (9B7ee9...CF2F). Transaction details include an amount of 0 ETH (0.00 USD), a gas limit of 841632 units, a gas price of 1 Gwei, and a max transaction fee of 0.000841 ETH (0.60 USD). The total cost is 0.000841 ETH (0.60 USD). The data included is 4 bytes. Three buttons are visible: 'RESET' (orange), 'SUBMIT' (green, highlighted with a purple box), and 'REJECT' (red).

In the background, the Remix IDE interface is visible, showing a code editor with Solidity code for a contract named 'AwardToken.sol'. The console at the bottom shows the following output:

```
> remix:batch
creation of AwardToken pending...
https://ropsten.etherscan.io/tx/0xef582524ea958e0b021dc796ac50ebecc16864c8b09443a94d7e1b8f349558f7
[block:3159186 txIndex:14] from:0x309...26e55 to:AwardToken.(constructor) value:0 wei
data:0x00...70029c0510...35817
transact to AwardToken.startRound pending ...
```

The last line of the console output, 'transact to AwardToken.startRound pending ...', is highlighted with a purple box.

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

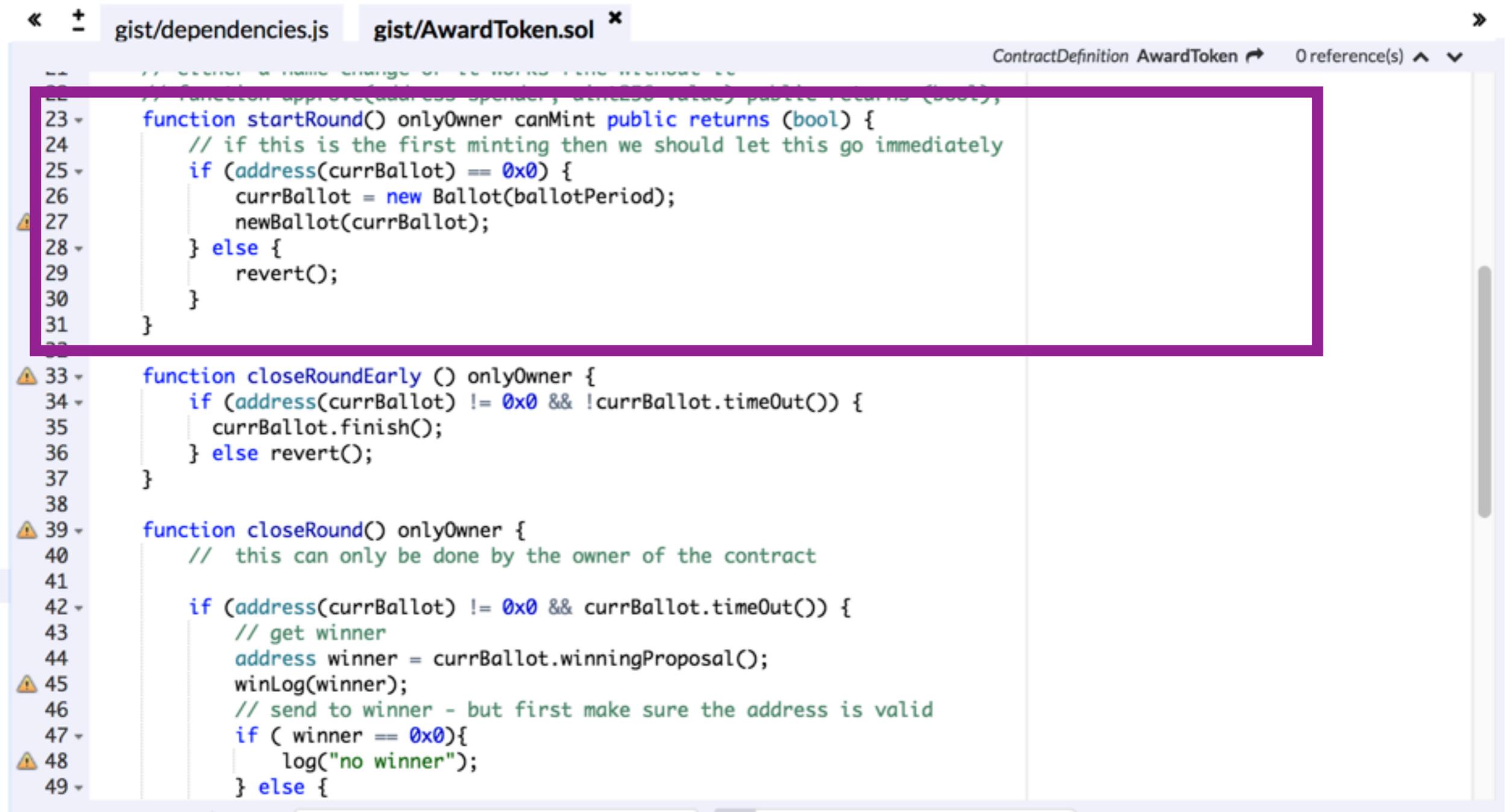
to see the logs

✓ [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

Checkout the startRound function in the editor



The image shows a code editor window with two tabs: 'gist/dependencies.js' and 'gist/AwardToken.sol'. The 'AwardToken.sol' tab is active, showing a Solidity contract definition. The function 'startRound()' is highlighted with a purple box. The code is as follows:

```
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

A screenshot of a blockchain explorer interface showing a list of functions for 'AwardToken at 0x9b7...0cf2f (blockchain)'. The 'currBallot' function is highlighted with a purple box. The interface includes a search bar at the top, a list of functions with their parameters, and a close button (X) in the top right corner.

Function	Parameters
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

AwardToken at 0x9b7...0cf2f (blockchain) 

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: 0xbE7bF60cee009DCDb2Ad8D045c19e76597bbF3c6	
getPreviousWinners		
mintingFinished		
owner		
prevWinners	uint256	▼
totalSupply		

Switch to Ballot

(which is loaded from AwardToken.sol)

Run tab: dropdown

The image shows a screenshot of a development environment's 'Run' tab. At the top, there are navigation tabs: 'Compile', 'Run' (which is active), 'Settings', 'Analysis', 'Debugger', and 'Support'. Below these tabs, there are several configuration fields:

- Environment:** A dropdown menu showing 'Injected Web3' with a small icon and 'Ropsten (3)' with a downward arrow and an information icon.
- Account:** A dropdown menu showing '0x309...26e55 (10.994338592 ether)' with a downward arrow and a plus icon.
- Gas limit:** A text input field containing '3000000'.
- Value:** A text input field containing '0' and a unit dropdown menu showing 'wei' with a downward arrow.

At the bottom of the interface, there is a large dropdown menu with a black border and a purple highlight. The dropdown is currently set to 'Ballot' and has a downward arrow on the right side.

Access Ballot contract

Paste address + click At Address

Compile Run Settings Analysis Debugger Support

Environment Injected Web3 Ropsten (3) ⓘ

Account 0x309...26e55 (10.994338592 ether) 📄 ⊕

Gas limit 3000000

Value 0 wei ▼

Ballot ▼

Deploy uint256 duration ▼

0xbE7bF60cee009DCDb2Ad8D045c19 At Address

See autogenerated UI

Interactive UI for Ballot.sol contract

Compile Run Settings Analysis Debugger Support

Environment Injected Web3 Ropsten (3) ⓘ

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

Gas limit 3000000

Value 0 wei

Ballot

Deploy uint256 duration

0xE7bF60cee009DCDb2Ad8D045c19 At Address

0 pending transactions

AwardToken at 0x9b7...0cf2f (blockchain) ⓘ

Ballot at 0xE7...bF3c6 (blockchain) ⓘ

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

Add a new proposal

Expand addProposal function

The screenshot shows a blockchain explorer interface for a contract named "Ballot at 0xbE7...bF3c6 (blockchain)". The interface displays a list of functions and variables. The "addProposal" function is highlighted in red, and its dropdown arrow is circled in purple. The "vote" function is also highlighted in red. The other functions and variables are highlighted in blue.

Function/Variable	Signature
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) ▼	
Account	0x309...26e55 (10.994338592 ether)		 
Gas limit	3000000		
Value	0	wei ▼	

Type a proposal

Run tab: Account

The screenshot shows a web interface for adding a proposal. At the top, there is a tab labeled "Ballot at 0xbE7...bF3c6 (blockchain)" with a close button (x) on the right. Below the tab, the function name "addProposal" is displayed. The form contains three input fields: "desc:" with the value "I think you could add a new feature to Remix that does...", "title:" with the value "This is my Remix improvements proposal", and "targetAddr:" with the value "address". A red "transact" button is located at the bottom right of the form.

▼ 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)**  ✕

addProposal ^

desc:

title:

targetAddr:

Execute addProposal

transact button

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

addProposal ^

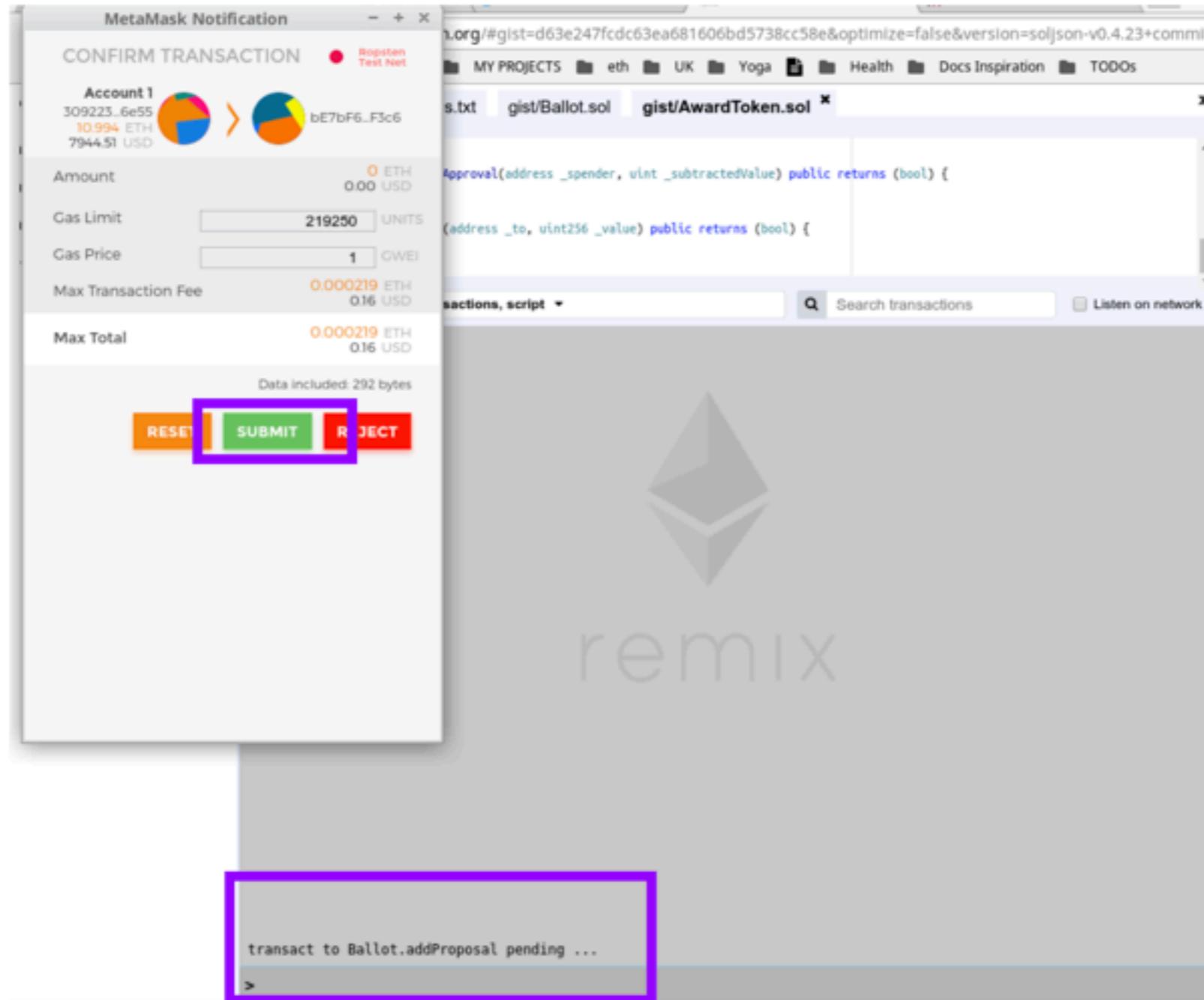
desc:

title:

targetAddr:

Confirm the transaction

Submit button



Execute getProposals

getProposals call

The screenshot shows a web interface for a blockchain contract named "Ballot at 0xbE7...bF3c6 (blockchain)". A list of functions is displayed, with "getProposals" highlighted by a purple box. The functions and their parameters are as follows:

Function Name	Parameters
addProposal	string desc, string title, address targetAddr
vote	address proposal
getProposals	
proposals	address
proposalsSender	uint256
timeOut	
winningProposal	

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

Vote for one Proposal

Paste Proposal Address you want to vote for

Compile Run Settings Analysis Debugger Support

Account 0x309...26e55 (10.994119342 ether)

Gas limit 3000000

Value 0 wei

Ballot

Deploy uint256 duration

0xbE7bF60cee009DCDb2Ad8D045c19 At Address

0 pending transactions

Ballot at 0xbE7...bF3c6 (blockchain)

addProposal string desc string title address targetAddr

vote 0x6Acd3829405CaFD677C7792c27a4c1c3013d7534

Execute vote transaction

vote button

Compile Run Settings Analysis Debugger Support

Account 0x309...26e55 (10.994119342 ether)

Gas limit 3000000

Value 0 wei

Ballot

Deploy uint256 duration

0xbE7bF60cee009DCDb2Ad8D045c19 At Address

0 pending transactions

Ballot at 0xbE7...bF3c6 (blockchain)

addProposal string desc, string title, address targetAddr

vote x6Acd3829405CaFD677C7792c27a4c1c3013d7534

Confirm the transaction

Submit button

The image shows a MetaMask transaction confirmation window overlaid on a web browser. The MetaMask window is titled "MetaMask Notification" and "CONFIRM TRANSACTION". It displays account information for "Account 1" (309223...6e55) and "Account 2" (bE7bF6...F3c6). Transaction details include: Amount: 0.00 ETH / 0.00 USD; Gas Limit: 64722 UNITS; Gas Price: 1 GWEI; Max Transaction Fee: 0.000064 ETH / 0.05 USD; Max Total: 0.000064 ETH / 0.05 USD. At the bottom, there are buttons for "RESE", "SUBMIT", "REJECT", and "REJECT ALL". The "SUBMIT" button is highlighted with a purple box.

The background browser window shows a web application with a code editor. The code includes a function `Approval(address _spender, uint _subtractedValue) public returns (bool) {` and another function `(address _to, uint256 _value) public returns (bool) {`. Below the code, there is a "Search transactions" field and a "Listen on network" checkbox. A transaction log is visible, showing a call to `Ballot.getProposals()` and a subsequent `transact to Ballot.vote pending ...` message, which is also highlighted with a purple box.

Check if tx succeeded

Terminal logs in Remix

The screenshot displays the Remix IDE interface. On the left, a file explorer shows a project structure with folders for 'browser', 'config', 'github', and 'gist'. Under 'gist', files 'AwardToken.sol', 'Ballot.sol', 'TUTORIAL.md', and 'dependencies.txt' are listed. The main editor window shows Solidity code for 'Ballot.sol' with lines 63-73 visible. The code includes functions 'decreaseApproval' and 'transfer', both of which call 'revert()'.

Below the code editor, the 'Transaction Log' panel is active, showing a search bar and a 'Listen on network' checkbox. It displays two transaction calls:

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

from	0x3092232fb25e6b359a9fead9ed07ad752ff26e55
to	Ballot.getProposals() 0xbE7bF60cee090Cdb2Ad80045c19e76597bbf3c6
input	0x62564c48
decoded input	{}
decoded output	{ "0": "address[]: 0x3092232fB25e6b359a9fEad9eD07Ad752Ff26e55, 0xFd0151afB685Cd8735AfE7685D21355589602b8c, 0x6AcD3829405CaFD677C7792c27a4c1c3013d7534" }
logs	[]

Below this, another transaction call is shown:

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

Following this, the log shows 'transact to Ballot.vote pending ...'. The final transaction log entry, highlighted with a red box, is:

- ▶ [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;  
    }  
}
```



```
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

(Click only one)

>	PROPOSAL 1 Lorem ipsum dolor sit amet, consectetur adipiscing... 1 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

The screenshot shows a web interface for a blockchain contract named "Ballot at 0x712...0Aa64 (blockchain)". It lists several methods: "addProposal" (with parameters "string desc, string title, address targetAddr"), "finish", "vote" (with parameter "address proposal"), and "getProposals". The "getProposals" method is selected, displaying a list of two proposals under the heading "0: address[]:". The first proposal is from address "0x9Ae59aF2E33480cAa48f2DC6F6CeDe7FFAb06Ff6,0xdc7b1AaC1D13d58C" and the second is from "EcEEc58434C1E32Fe2A1297f".

Method	Parameters
addProposal	string desc, string title, address targetAddr
finish	
vote	address proposal
getProposals	

0: address[]:

- 0x9Ae59aF2E33480cAa48f2DC6F6CeDe7FFAb06Ff6,0xdc7b1AaC1D13d58C
- EcEEc58434C1E32Fe2A1297f

2 proposals have been added

@ninabreznik @ryestew @yann300 @serapath @iurimatias

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

bit.ly/remix-workshop-repository

Current AwardToken

0x0933e16e31e3f7cbbdcd2375c7b9a28fab7351fe