

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

The screenshot displays a web interface for a ballot Dapp. At the top right, there is a button labeled "CREATE NEW PROPOSAL". Below this, four buttons represent the voting rounds: "ROUND: 1", "ROUND: 2", "ROUND: 3", and "ROUND: 4". A central banner reads "Vote for proposal and help us reward the projects that benefit the community!". The main section is titled "PROPOSALS" and includes a sub-header "PROPOSAL TITLE/DESCRIPTION" and a note "Choose only one". Two proposals are visible: "PROPOSAL 1" with a description "Lorem ipsum dolor sit amet, consectetur adipiscing..." and "1 vote(s)", and "PROPOSAL 3" with a description "Neque porro quisquam est, qui dolorem ipsum quia ..." and "0 vote(s)". Each proposal has a right-pointing arrow on the left and a radio button on the right.

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 is also a "RATINGS" section with star icons. 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 several transactions. Below the preview, there is a "RELATED" section with other extensions like "MyEtherWallet", "Disable Extensions Temporarily", "EtherAddressLookup", and "The FFZ Add-On Pack". The page also includes a "COMPATIBLE WITH YOUR DEVICE" section and a "DESCRIPTION" section that explains that MetaMask is an Ethereum browser extension that injects the Ethereum web3 API into every website's javascript context. The page is viewed in a browser window with the address bar showing "chrome.google.com/webstore".

chrome web store

metamask

MetaMask

offered by <https://metamask.io>

★★★★★ (983) | Productivity | 1,082,276 users

ADDED TO CHROME

OVERVIEW | REVIEWS | SUPPORT | RELATED

Compatible with your device

Ethereum Browser Extension

MetaMask is an extension for accessing Ethereum enabled distributed applications, or "Dapps" in your normal Chrome browser!

The extension injects the Ethereum web3 API into every website's javascript context, so that dapps can read from the blockchain.

MetaMask also lets the user create and manage their own identities, so when a Dapp wants to perform a transaction and write to the blockchain, the user gets a secure interface to review the transaction, before answering or rejecting it.

Website | Report Abuse

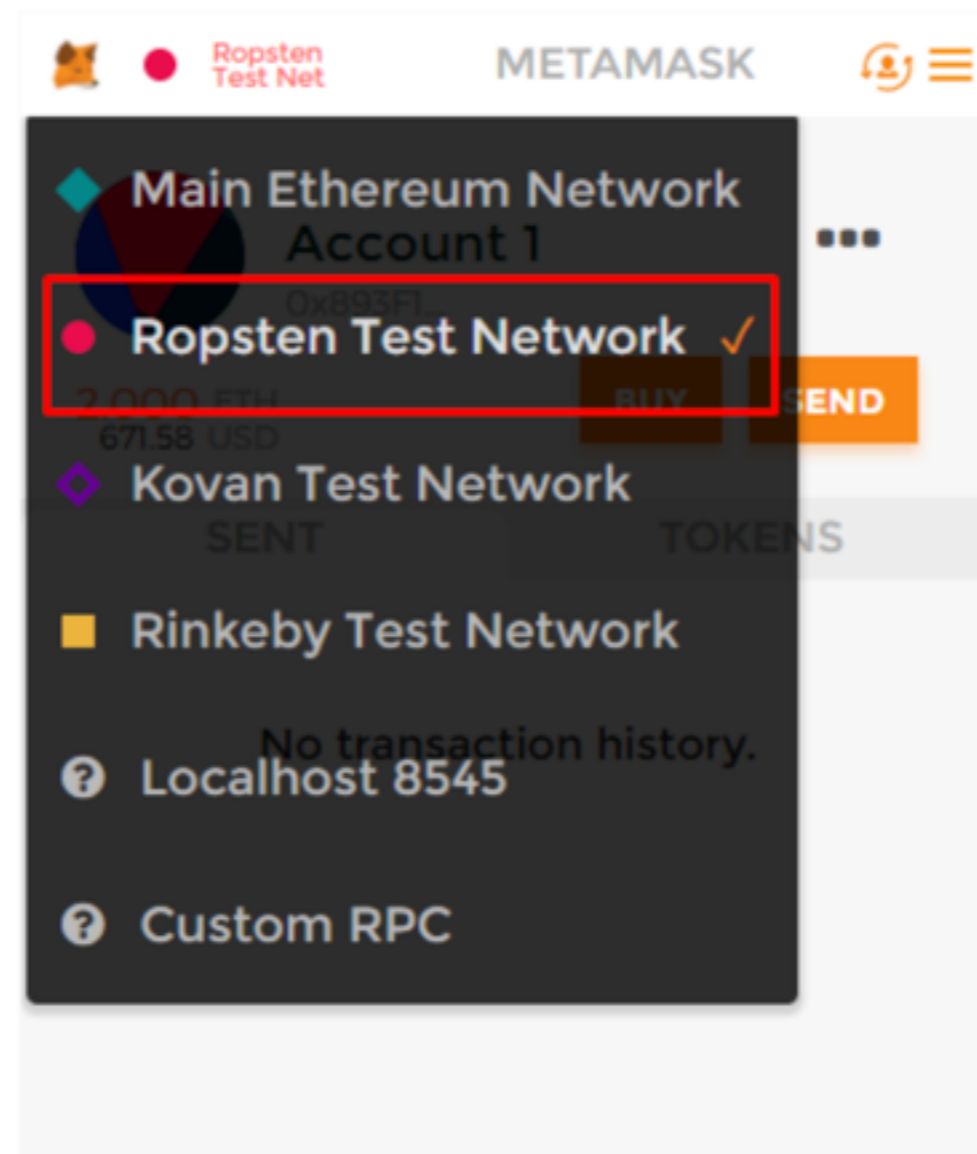
Additional Information

Version: 4.6.1
Updated: April 30, 2018
Size: 5.58MiB
Languages: See all 16

MyEtherWallet (260) | Disable Extensions Temporarily (250) | EtherAddressLookup (65) | The FFZ Add-On Pack (117)

Login to Metamask

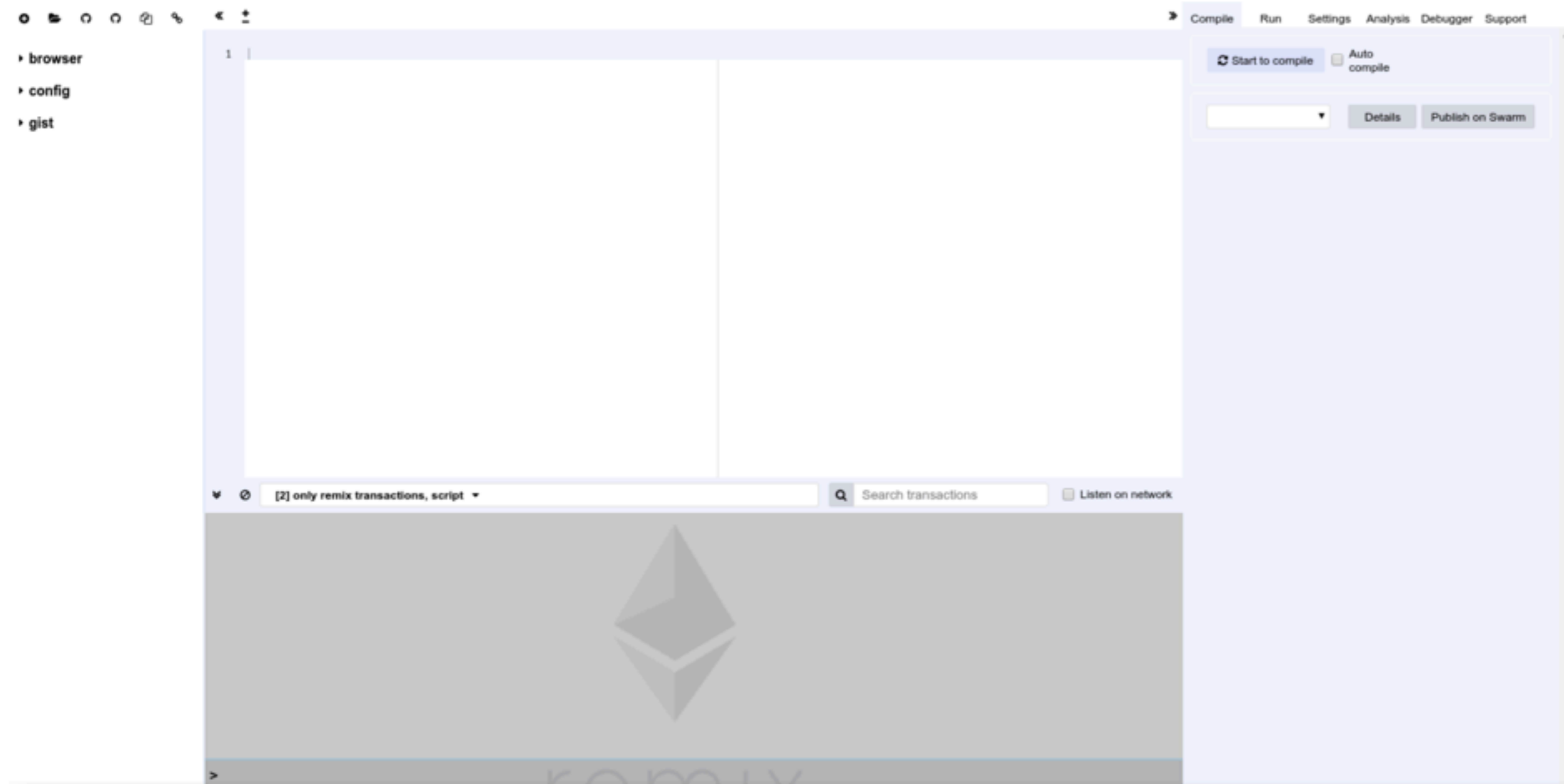
Ropsten Test Network



Let's get started

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

<https://remix-alpha.ethereum.org>



Remix Tour

File Explorer

<https://remix-alpha.ethereum.org>

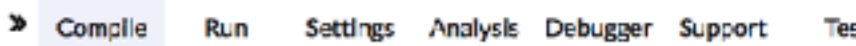
Compile Tab (active)



- 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 with two main panels. The left panel displays the source code for a contract named 'AwardToken'. The right panel shows the 'Run' configuration options.

```
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
```

The 'Run' configuration panel on the right includes the following settings:

- Environment:** Injected Web3 (Ropsten (3))
- Account:** 0x667...d091d (1.453587112999635446)
- Gas limit:** 3000000
- Value:** 0 wei

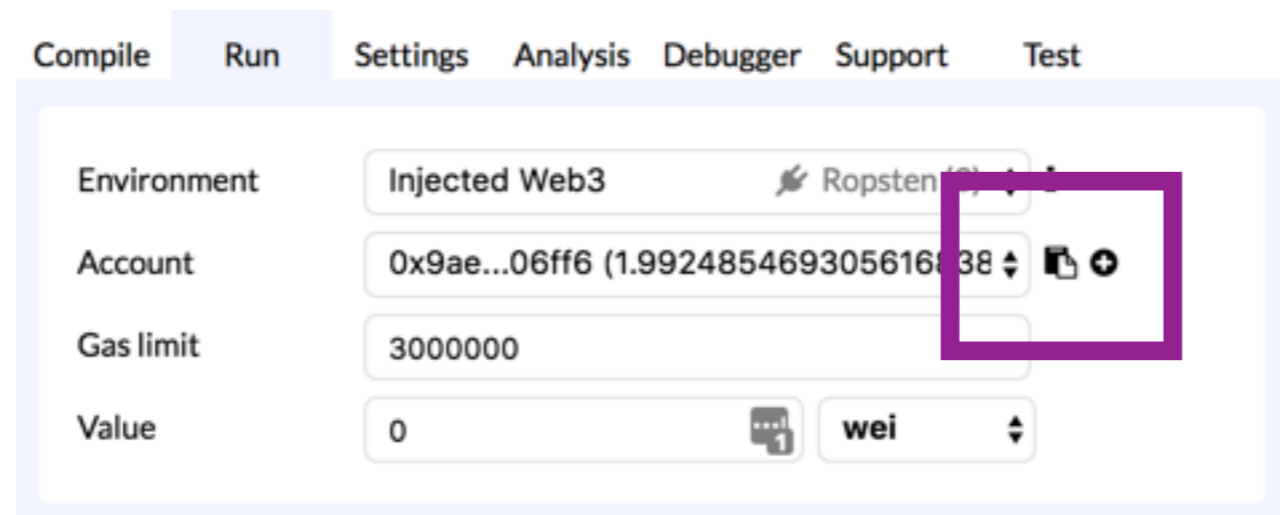
Below these settings, there is a dropdown menu showing 'AwardToken', a 'Deploy' button, and a 'Load contract from Address' button with an 'At Address' button next to it.

Get some TEST ether

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

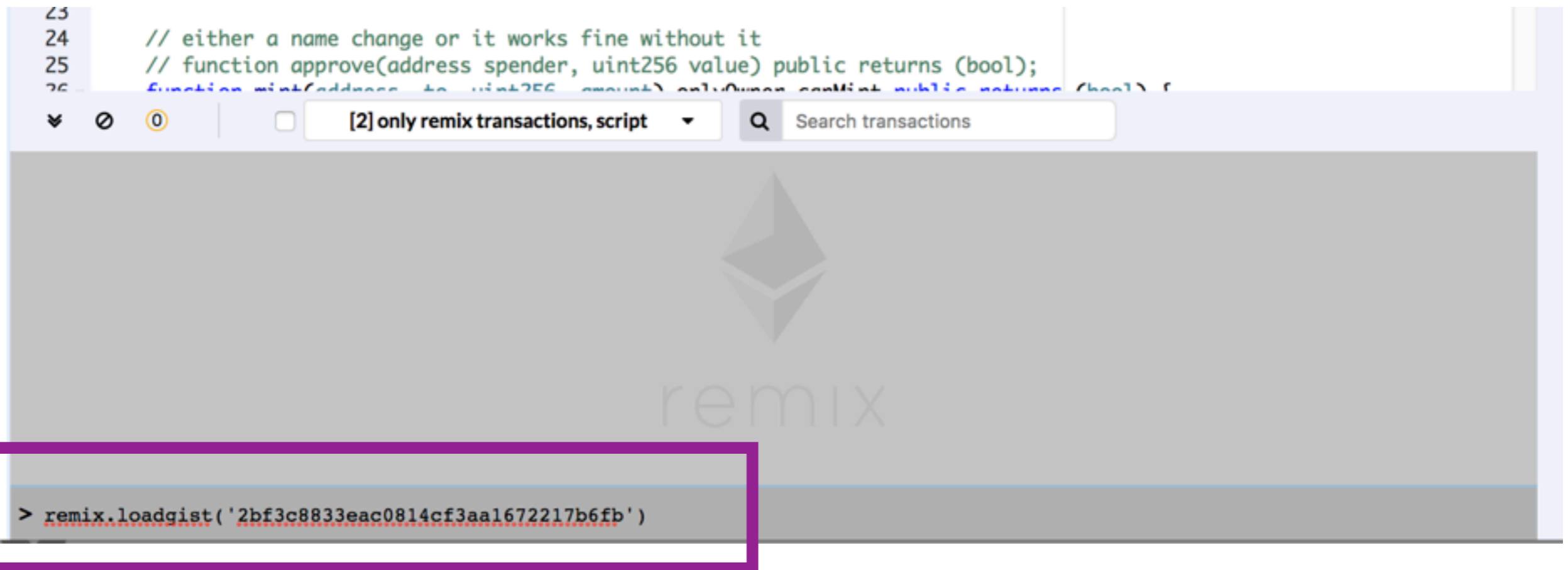
But FIRST:

Copy your address - here or in Metamask



Load files to Remix

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



here in the console

Open file

gist/dependencies.js



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

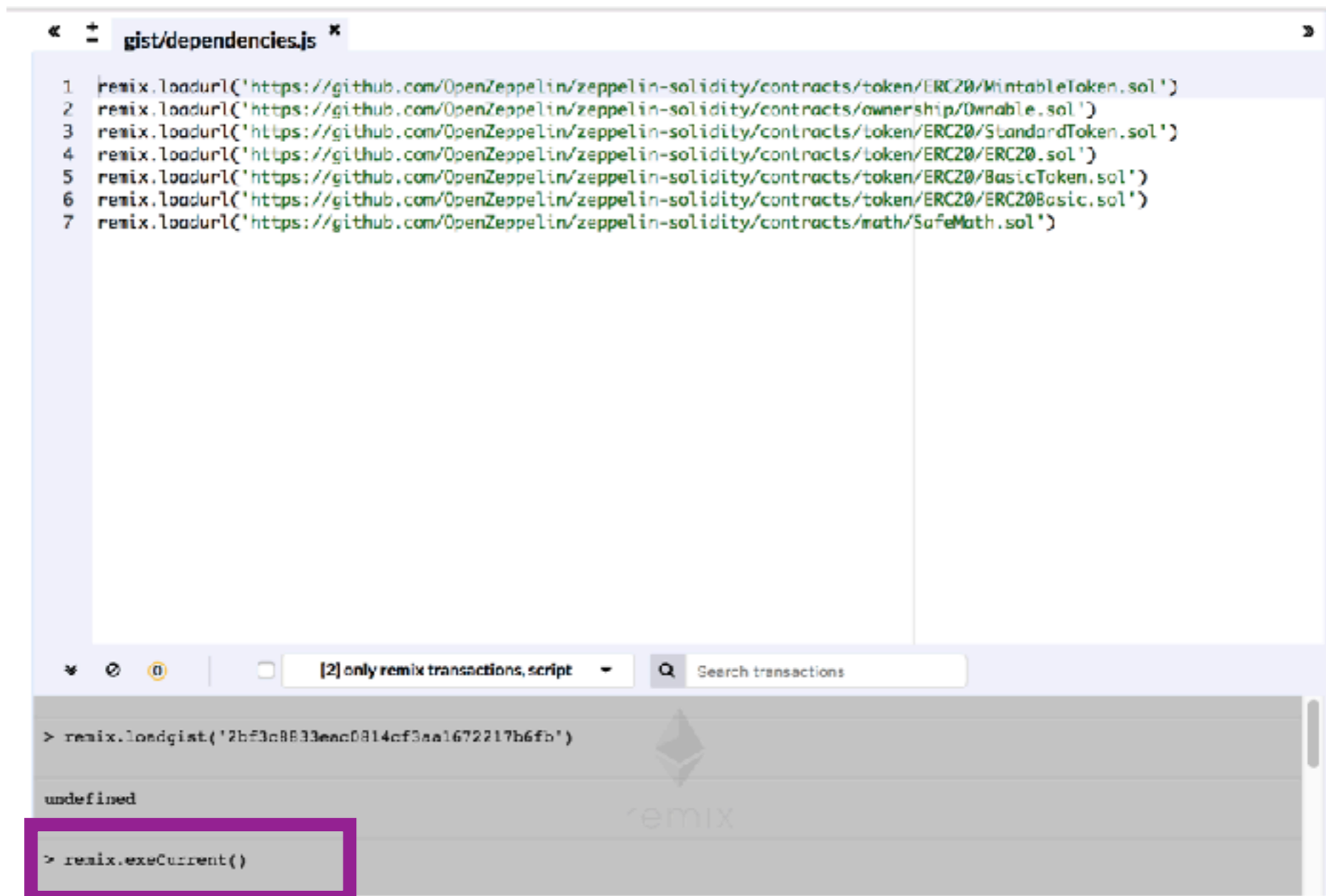
« + gist/dependencies.js x

```
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/contracts



```
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

```
> remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/MintableToken.sol')
remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/ownership/Ownable.sol')
remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/StandardToken.sol')
remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/ERC20.sol')
remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/BasicToken.sol')
remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/token/ERC20/ERC20Basic.sol')
remix.loadurl('https://github.com/OpenZeppelin/zeppelin-solidity/contracts/math/SafeMath.sol')
```


Open file

gist/AwardToken



- ▶ browser
- ▶ config
- ▼ github
 - ▼ OpenZeppelin
 - ▼ zepppelin-solidity
 - ▼ contracts
 - ▼ ownership
 - Ownable.sol
 - ▼ token
 - ▶ ERC20
 - ▼ math
 - SafeMath.sol
 - ▼ gist
 - AwardToken.sol
 - Ballot.sol
 - README.md
 - dependencies.js

```
gist/dependencies.js  gist/AwardToken.sol x
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         if (address(currBallot) == 0x0) {
```

[2] only remix transactions, script

Search transactions

Compile the contract

Compile tab: Start to compile button

The image shows the Remix IDE interface. On the left, the 'gist/AwardToken.sol' file is open, displaying Solidity code for an 'AwardToken' contract. The code includes imports for 'MintableToken' and 'Ballot', and defines a constructor and a 'getPreviousWinners' function. On the right, the 'Compile' tab is active, showing a 'Start to compile' button highlighted with a purple box. Other buttons include 'Auto compile' and 'Hide warnings'. Below the buttons, the contract name 'AwardToken' is displayed, along with options for 'Details', 'Publish on Swarm', 'ABI', and 'Bytecode'. A warning message states: 'Static Analysis raised 38 warning(s) that requires your attention. Click here to show the warning(s)'. Below this, two specific warnings are shown: 'Warning: Defining constructors as function' for 'Ballot' and 'AwardToken'.

```
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        if (address(currBallot) == 0x0) {
```


See compiled contracts

AwardToken + all dependencies



Imported Contracts

The screenshot shows a web3 development interface with a top navigation bar containing: > Compile, Run, Settings, Analysis, Debugger, Support, and Test. The 'Run' tab is active. Below the navigation bar, there are four input fields for transaction settings: Environment (Injected Web3), Account (0x9ae...06ff6 (1.992485469305616838)), Gas limit (3000000), and Value (0 wei). Below these fields is a scrollable list of imported contracts: AwardToken (checked), Ballot, SafeMath, Ownable, BasicToken, ERC20, ERC20Basic, MintableToken, and StandardToken. At the bottom, there is a section labeled 'Deployed Contracts' with a trash icon.

Environment: Injected Web3 Ropsten (3) ⓘ

Account: 0x9ae...06ff6 (1.992485469305616838) ⓘ ⊕

Gas limit: 3000000

Value: 0 wei ⓘ

- ✓ AwardToken
- Ballot
- SafeMath
- Ownable
- BasicToken
- ERC20
- ERC20Basic
- MintableToken
- StandardToken

Deployed Contracts ⓘ

Deploy the contract

Run tab: Deploy button

The image shows a screenshot of the 'Run' tab in a development tool. The interface includes a navigation bar with tabs for 'Compile', 'Run', 'Settings', 'Analysis', 'Debugger', and 'Su'. Below the navigation bar, there are several configuration fields: 'Environment' set to 'Injected Web3' with a network selection icon and 'Ropsten (3)'; 'Account' set to '0x309...26e55 (11 ether)' with a dropdown arrow and icons for copying and adding; 'Gas limit' set to '3000000'; and 'Value' set to '0' with a unit selector set to 'wei'. At the bottom, a dropdown menu shows the contract name 'AwardToken'. A red rectangular box highlights the 'Deploy' button located below the contract name dropdown.

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 composite screenshot of a web browser interface. On the left, a MetaMask notification window is open, titled "CONFIRM TRANSACTION". It displays account information for "Account 1" (309223...6e55, 11,000 ETH, 7921.32 USD) and a "New Contract" button. Below this, there are input fields for "Amount" (0 ETH, 0.00 USD), "Gas Limit" (2409888 UNITS), "Gas Price" (1 CWei), "Max Transaction Fee" (0.002409 ETH, 1.73 USD), and "Max Total" (0.002409 ETH, 1.73 USD). At the bottom of the notification are three buttons: "RESET" (orange), "SUBMIT" (green, highlighted with a purple box), and "REJECT" (red). The background shows the Remix IDE interface. The top navigation bar includes "Compile", "Run", "Settings", "Analysis", "Debugger", and "Support". The main editor area displays Solidity code for an "AwardToken" contract, which inherits from "MintableToken". The code includes a constructor for "AwardToken" with a quantity of 100, and a "getPreviousWinners" function. Below the code is a search bar for transactions. On the right side, there is a control panel with "Environment" set to "Injected Web3" (Ropsten (3)), "Account" set to "0x309...26e55 (11 ether)", "Gas limit" set to "3000000", and "Value" set to "0" (wei). A dropdown menu shows "AwardToken" and a "Deploy" button is visible. Below the control panel, there is a section for "1 pending transactions" and "0 contract instances". At the bottom, a terminal window 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 shows a web3 development tool interface with the following components:

- Navigation:** Compile, Run (selected), Settings, Analysis, Debugger, Support, Test.
- Environment:** Injected Web3, Ropsten (3)
- Account:** 0x9ae...06ff6 (1.993121706305616838)
- Gas limit:** 3000000
- Value:** 0, wei
- Contract Selection:** AwardToken (selected), Deploy button, Load contract from Address, At Address button.
- Transactions:** Transactions recorded: 1
- Deployed Contracts:** AwardToken at 0x574...40360 (blockchain). A purple box highlights the right-pointing arrow icon next to the contract name.

Behold!

The Interactive UI for AwardToken.sol contract

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		
getPreviousWinners		
mintingFinished		
owner		
prevWinners	uint256	⌵
totalSupply		

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 dialog overlaid on a Remix IDE interface. The dialog is titled "MetaMask Notification" and "CONFIRM TRANSACTION" on the "Ropsten Test Net". It displays account information for "Account 1" (309223...6e55) and the recipient (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). Data included is 4 bytes. At the bottom of the dialog are three buttons: "RESET" (orange), "SUBMIT" (green, highlighted with a purple box), and "REJECT" (red).

The background shows the Remix IDE with a code editor displaying Solidity code for a contract named "AwardToken". 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 "transact to AwardToken.startRound pending ..." line 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

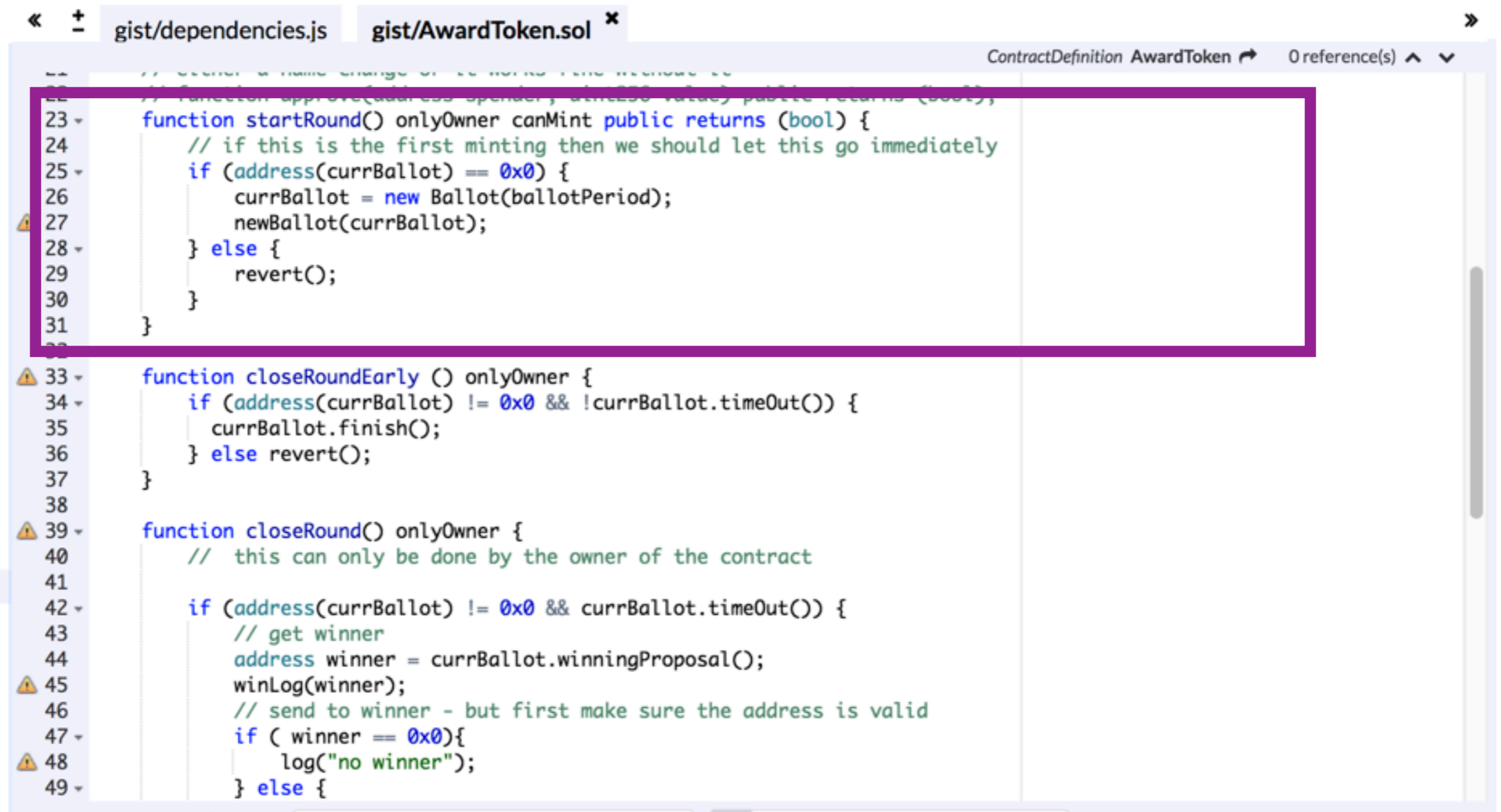
to see the logs

✓ [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



```
ContractDefinition AwardToken 0 reference(s) ^ v
--
// 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

Run tab: dropdown

Compile **Run** Settings Analysis Debugger Support

Environment  Ropsten (3) ▼ 

Account ▼  

Gas limit

Value ▼

Ballot ▼

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

The screenshot displays the interface of a development tool, likely Remix, for interacting with a smart contract. The top navigation bar includes 'Compile', 'Run', 'Settings', 'Analysis', 'Debugger', and 'Support'. The 'Run' tab is active, showing the following configuration:

- Environment: Injected Web3 (Ropsten (3))
- Account: 0x309...26e55 (10.994338592 ether)
- Gas limit: 3000000
- Value: 0 wei

The contract 'Ballot' is selected, and the 'Deploy' button is highlighted. The deployment target address is 0xbE7bF60cee009DCDb2Ad8D045c19. Below this, a section indicates '0 pending transactions'. A dropdown menu shows the selected contract: 'Ballot at 0xbE7...bF3c6 (blockchain)'. The methods available for interaction are listed below:

Method	Parameters
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

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:

transact

Execute addProposal

transact button

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

addProposal ^

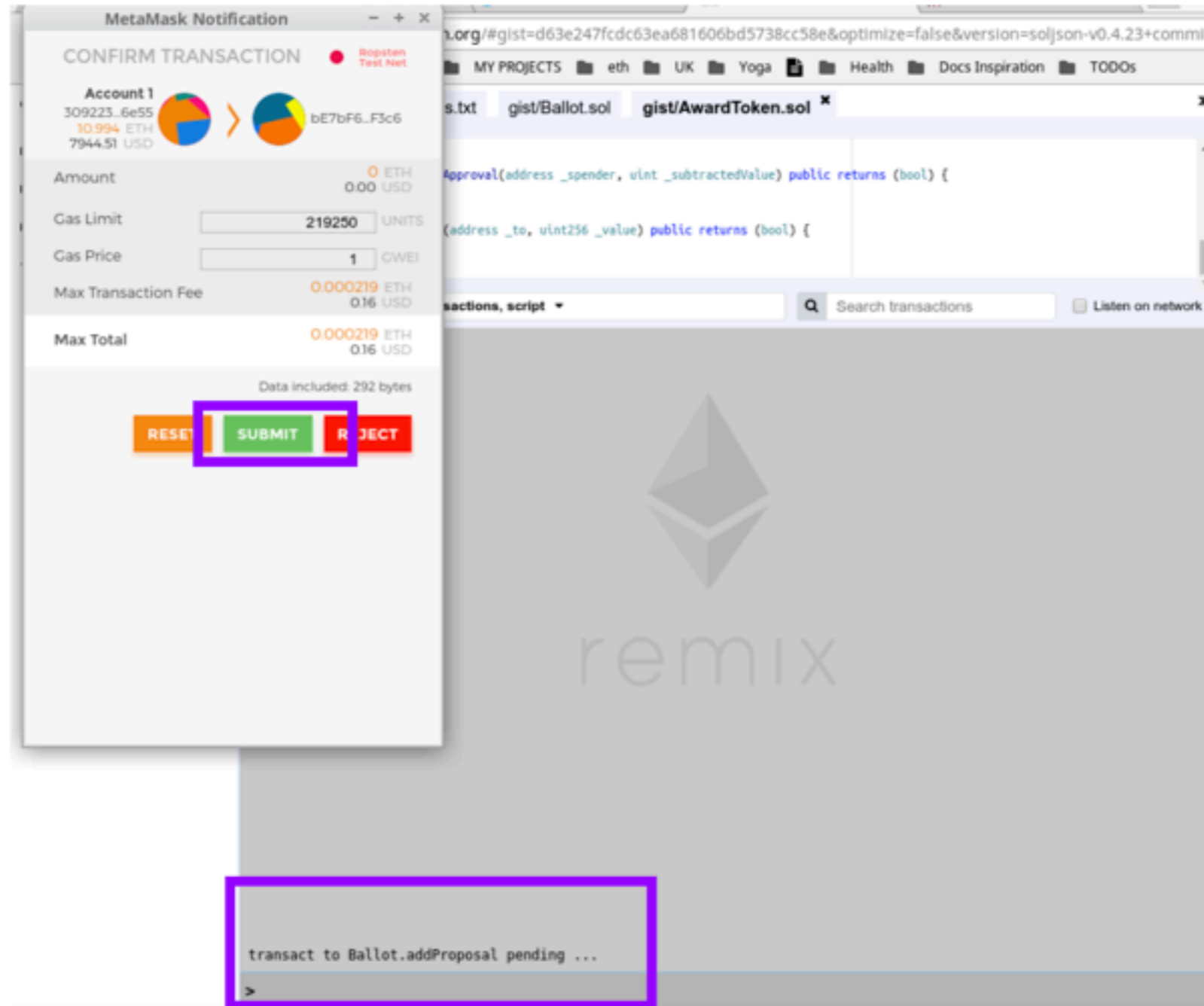
desc:

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) [copy] [refresh]

Gas limit 3000000

Value 0 wei [dropdown]

Ballot [dropdown]

Deploy uint256 duration [dropdown]

0xbE7bF60cee009DCDb2Ad8D045c19 At Address

0 pending transactions [copy] [play] [trash]

Ballot at 0xbE7...bF3c6 (blockchain) [copy] [close]

addProposal string desc string title address targetAddr [dropdown]

vote 0x6Acd3829405CaFD677C7792c27a4c1c3013d7534 [dropdown]

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 dialog overlaid on a web browser. The dialog 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 of the dialog, there are four buttons: "RESE" (orange), "SUBMIT" (green, highlighted with a purple box), "REJECT" (red), and "REJECT ALL" (red). The background browser window shows a web page with a search bar and a "Listen on network" checkbox. Below the search bar, there is a "Debug" section with a table of transaction logs. The log entry for "transact to Ballot.vote pending ..." is highlighted with a purple box.

MetaMask Notification

CONFIRM TRANSACTION

Account 1
309223...6e55
10.994 ETH
7865.63 USD

Account 2
bE7bF6...F3c6

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

Data included: 36 bytes

RESE SUBMIT REJECT

REJECT ALL

Search transactions

Listen on network

Debug

0x3092232fb25e6b359a9fead9ed07ad752ff26e55
Ballot.getProposals() 0xbE7bF60cee0990CDb2Ad8D045c19e76597bbF3c6
0x62564c48
()
{ "0": "address[]: 0x3092232fb25e6b359a9fead9ed07ad752ff26e55, 0xFd0f51afB685Ccd8735Afe7685D21355589602b8c, 0x6AcD3829485CaFD677C7792c27a4c1c3013d7534" }
logs

call to Ballot.proposals

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

transact to Ballot.vote pending ...

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 shows Solidity code for 'Ballot.sol' with lines 63-73. The code includes functions 'decreaseApproval' and 'transfer', both of which call 'revert()'.

Below the code editor, the 'Debug Console' is open, showing transaction logs. The logs include:

- call to Ballot.getProposals
- call to Ballot.proposals
- transact to Ballot.vote pending ...
- A transaction log for block 3159861, txIndex 27, which is highlighted with a red box. The log shows: `[block:3159861 txIndex:27] from:0x309...26e55 to:Ballot.vote(address) 0xbe7...bf3c6 value:0 wei data:0x6dd...d7534 logs:0 hash:0xe0d...6c6eb`

from	0x3092232fb25e6b359a9fead9ed07ad752ff26e55
to	Ballot.getProposals() 0xbE7bF60cee090Cdb2Ad80045c19e76597bbf3c6
input	0x62564c48
decoded input	{}
decoded output	{ "0": "address[]: 0x3092232fb25e6b359a9fead9ed07ad752ff26e55, 0xfD0151afb685Cd8735Afe7685D21355589602b8c, 0x6AcD3829405CaFD677C7792c27a4c1c3013d7534" }
logs	[]

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

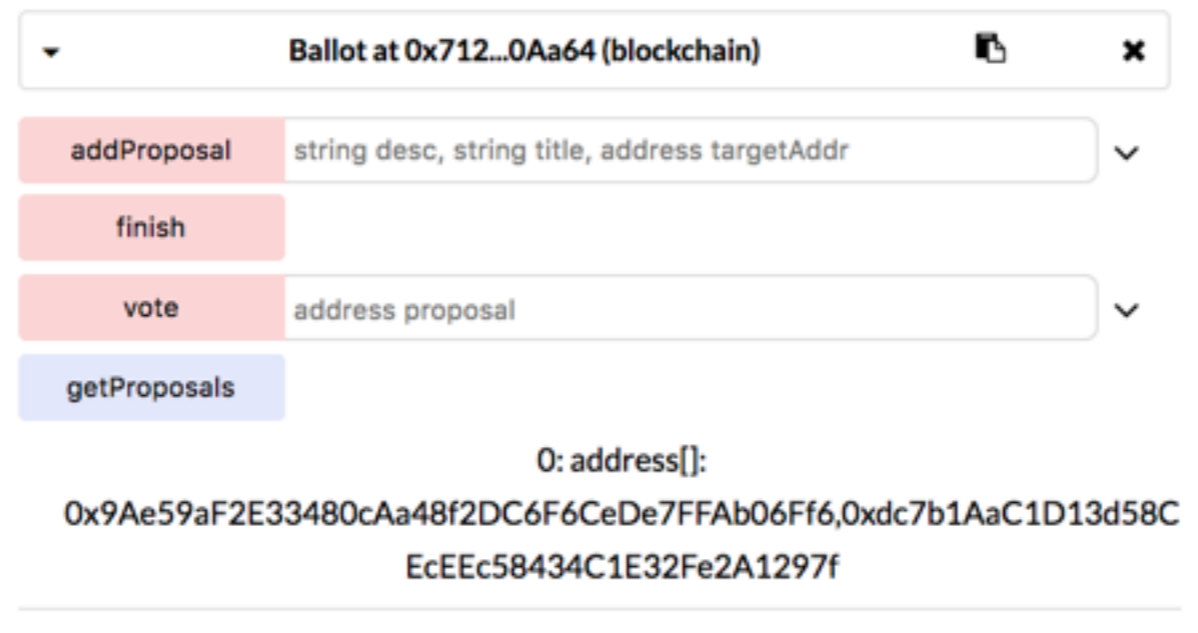
(Click only one)

>	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 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, showing a list of proposals. The first proposal is at index 0, with the parameter "address[]" containing two addresses: "0x9Ae59aF2E33480cAa48f2DC6F6CeDe7FFAb06Ff6,0xdc7b1AaC1D13d58CEcEEc58434C1E32Fe2A1297f".

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

2 proposals have been added

@ninabreznik @ryestew @yann300 @serapath @iurimatias

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