

Multiple ways of loading files intro to Remix

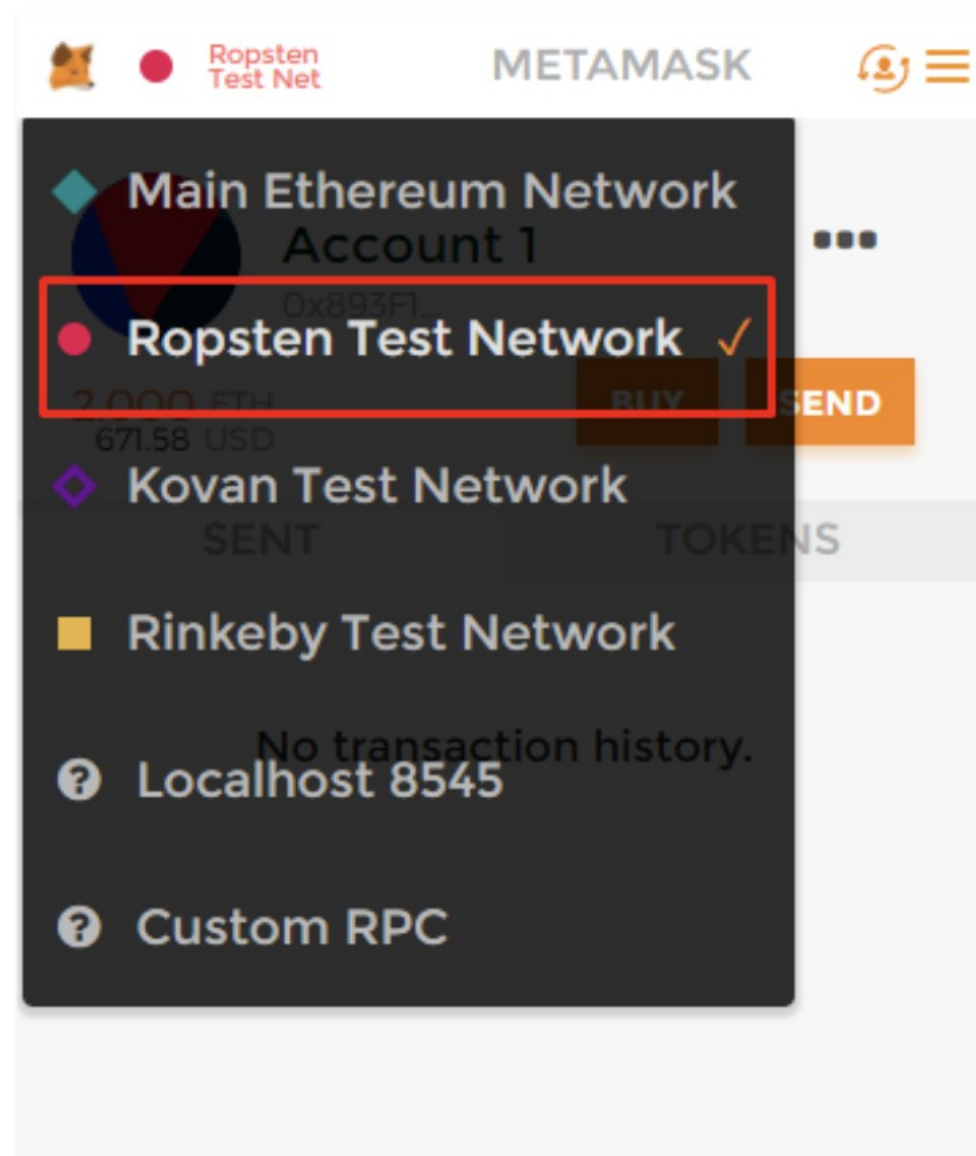
This tutorial covers the same info as the basic deployment tutorial – but it teaches you 3 different routes of getting access to the files we want to import:

- Through using remixd – which allows remix to look into a specific folder (that you specify) on your hard drive. In this tutorial we will use this along with using NPM
- Using the remix console to download files in a github repo
- Using the Zeppelin plugin for remix, that imports the contracts through https

This tutorial assumes some knowledge of Remix.

At the end of this tutorial, you could, if you fancy, continue to part 2 of the basic deployment tutorial.

Login to Metamask to choose Ropsten



Remix Tour

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

File Explorer

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

« + browser/Ballot_orig.sol x

```
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

» Compile Run Settings Analysis Debugger Support T

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 function Ballot(uint8 _numProposals) public + ^ (Relevant source part starts here and spans

0 [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
```

Terminal

>

Console

Remix Commands

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

The screenshot displays the Remix IDE interface. On the left, a file explorer shows a project named 'browser' containing files like 'AwardToken.sol', 'Ballot2.sol', 'Ballot_orig.sol', 'Donation.sol', 'README.md', 'multiSig2.sol', 'multisig.sol', 'multisig1.sol', 'scenario.json', and 'setup.txt'. The main editor shows a Solidity contract named 'Ballot' with the following code:

```
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
```

On the right, the 'Compile' tab is active, showing a 'Start to compile' button, 'Auto compile' checked, and 'Hide warnings' unchecked. Below this, a dropdown menu shows 'Ballot' selected, with buttons for 'Details', 'Publish on Swarm', 'ABI', and 'Bytecode'. A warning message states: 'Static Analysis raised 2 warning(s) that requires your attention. Click here to show the warning(s)'. Below this, a specific warning is shown: '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 following text:

```
- 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:

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

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

Web3 Provider

Choose: **Injected web3 (Ropsten)**

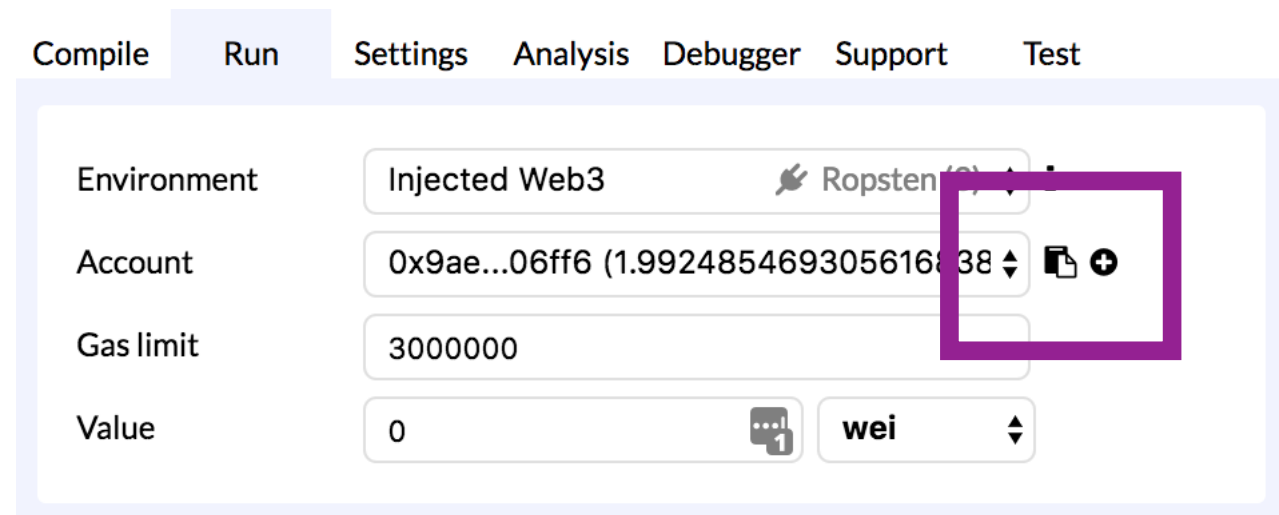
The screenshot shows the Remix IDE interface. On the left, the 'gist/AwardToken.sol' file is open, displaying Solidity code for an ERC20 token contract. The code includes imports for 'zeppelin-solidity/contracts/token/ERC20/MintableToken.sol' and 'gist/Ballot.sol', followed by a 'contract AwardToken is MintableToken' definition with various functions like 'AwardToken()', 'getPreviousWinners()', and 'startRound()'. On the right, the 'Run' tab is active, showing the 'Environment' settings. A purple box highlights the 'Environment' section, which includes a dropdown menu set to 'Injected Web3' and a sub-menu showing 'Ropsten (3)'. Below this, the 'Account' field displays '0x667...d091d (1.453587112999635446)', the 'Gas limit' is set to '3000000', and the 'Value' is set to '0' with a unit dropdown set to 'wei'. At the bottom of the Run tab, there is a dropdown menu showing 'AwardToken', a 'Deploy' button, and a section for 'Load contract from Address' with an 'At Address' button.

Get some TEST ether

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

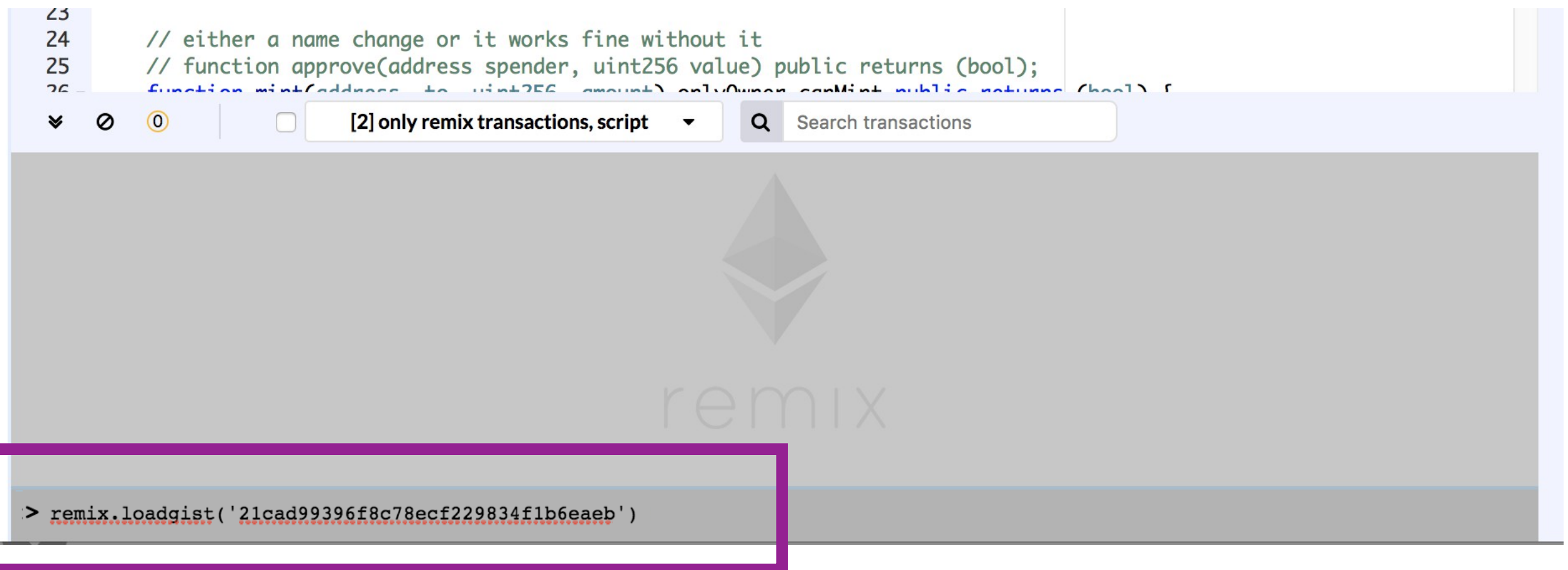
But FIRST:

Copy your address - here or in Metamask



Load files from a GIST

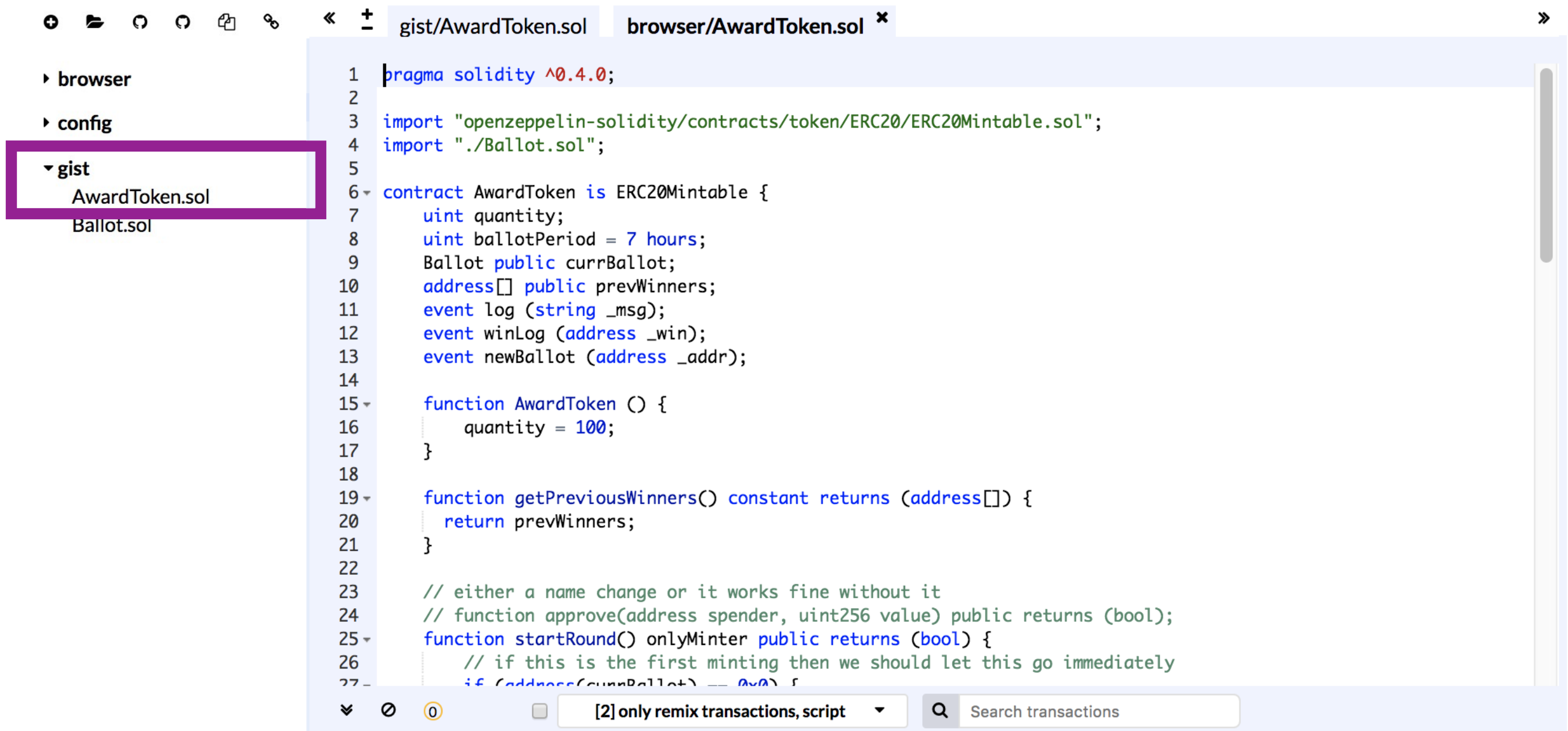
```
remix.loadgist('21cad99396f8c78ecf229834f1b6eaeb')
```



here in the console

Open file

gist/AwardToken



Setup for remixd:

To access local files on your computer
and npm modules

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

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

Type these commands:

npm init -y

npm install -g remixd (to install it globally)

npm install -E openzeppelin-solidity

(the npm module openzeppelin-solidity contains the contracts that we need for
awardToken.sol)

Get Remixd Running!

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

Type:

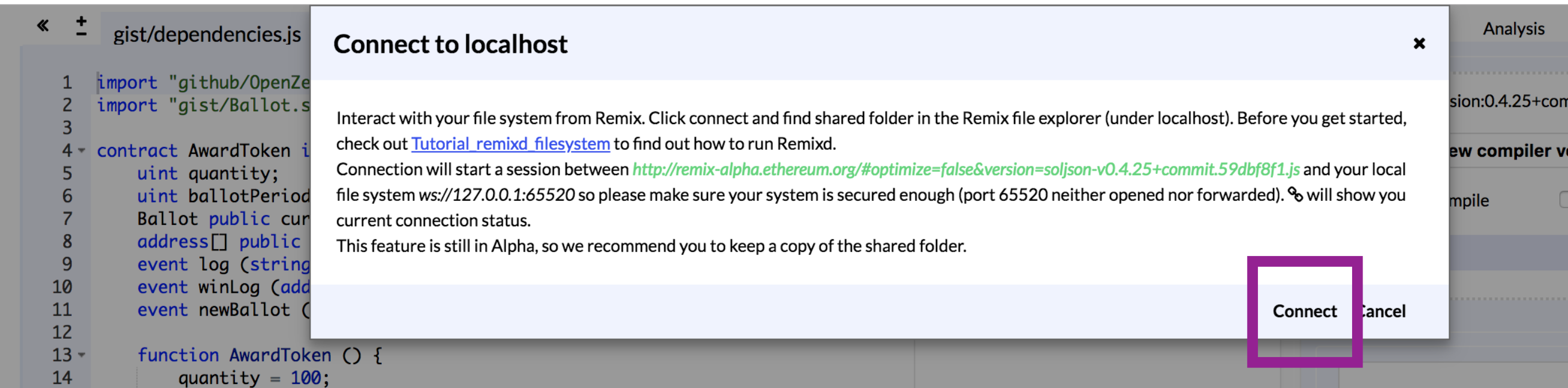
```
remixd -s ./ --remix-ide https://remix.ethereum.org
```

(or <http://remix-alpha.ethereum.org> or what ever version of remix you are on)

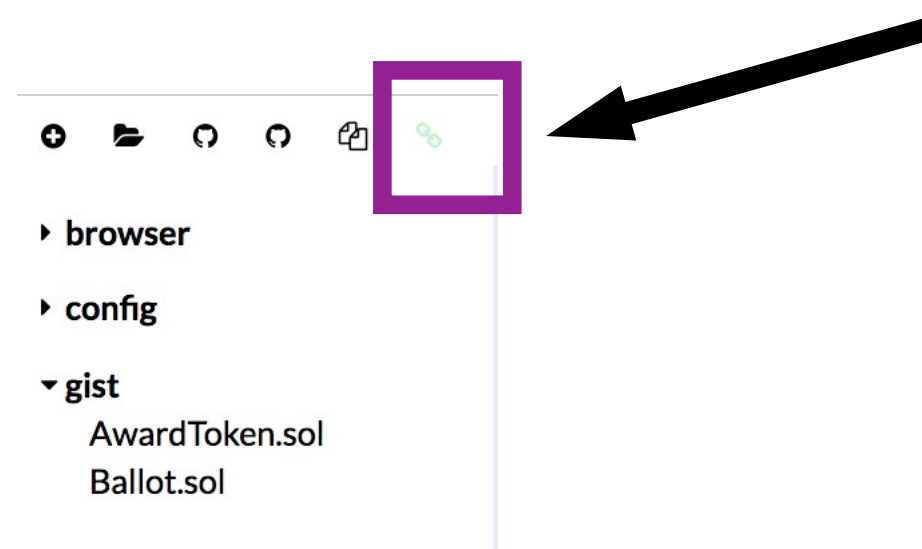
Then go to remix and turn on remixd by clicking the link icon at the top of the File Explorer (top left of Remix)



Click 'Connect'



Then the link icon should turn light green.



Compile AwardToken

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

Click the “Start to compile” button

The screenshot displays the Remix IDE interface. On the left, a file explorer shows the project structure with files like AwardToken.json, AwardToken.sol, and Ballot.sol. The main editor window shows the source code of AwardToken.sol, which includes Solidity code for a contract inheriting from ERC20Mintable. The code defines variables for quantity and ballot period, and includes functions for minting and retrieving winners. On the right, the 'Compile' tab is active, showing the current compiler version (0.4.25+commit.59dbf8f1.Emscripten.clang) and a 'Start to compile (Ctrl-S)' button, which is highlighted with a red rectangle. Below the compiler settings, there are options for 'Auto compile' and 'Hide warnings'. At the bottom of the right panel, a warning message states: 'Static Analysis raised 46 warning(s) that requires your attention. Click here to show the warning(s)'. Below this, a specific warning is shown: 'browser/Ballot.sol:26:5: Warning: Defining constructors as functions. function Ballot(uint duration) public { ... }'.

```
1 pragma solidity ^0.4.0;
2
3 import "openzeppelin-solidity/contracts/token/ERC20/ERC20Mintable.sol";
4 import "../Ballot.sol";
5
6 contract AwardToken is ERC20Mintable {
7     uint quantity;
8     uint ballotPeriod = 7 hours;
9     Ballot public currBallot;
10    address[] public prevWinners;
11    event log (string _msg);
12    event winLog (address _win);
13    event newBallot (address _addr);
14
15    function AwardToken () {
16        quantity = 100;
17    }
18
19    function getPreviousWinners() constant returns (address[]) {
20        return prevWinners;
21    }
22
23    // either a name change or it works fine without it
24    function approve(address spender, uint256 value) public returns (bool);
25    function startRound() onlyMinter public returns (bool) {
26        // if this is the first minting then we should let this go immediately
27        if (address(currBallot) == 0x0) {
```

If you have no compilation errors (in pink) then...

SUCCESS!!

Compilation errors look like this:

Compile is now orange

Error is pink

The screenshot shows the Remix IDE interface. At the top, there is a navigation bar with tabs: **Compile** (highlighted in orange), Run, Analysis, Testing, Debugger, Settings, and Support. Below the navigation bar, the main area is divided into sections. The first section, titled 'Current version:0.4.25+commit.59dbf8f1.Emscripten.clang', contains a dropdown menu labeled 'Select new compiler version', three checkboxes ('Auto compile' is checked, 'Enable Optimization' and 'Hide warnings' are unchecked), and a button labeled 'Start to compile (Ctrl-S)'. The second section contains a dropdown menu, a 'Swarm' button, and three buttons labeled 'Details', 'ABI', and 'Bytecode'. The third section, which is highlighted in pink, displays a compilation error: 'gist/AwardToken.sol:15:5: ParserError: Expected identifier but got'. Below the error message, the code snippet 'function AwardToken () {' is shown with a caret under the opening brace and a dashed line indicating the error location.

Compile

Run Analysis Testing Debugger Settings Support

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

Select new compiler version

☒ Auto compile ☐ Enable Optimization ☐ Hide warnings

Start to compile (Ctrl-S)

Swarm

Details ABI Bytecode

gist/AwardToken.sol:15:5: ParserError: Expected identifier but got

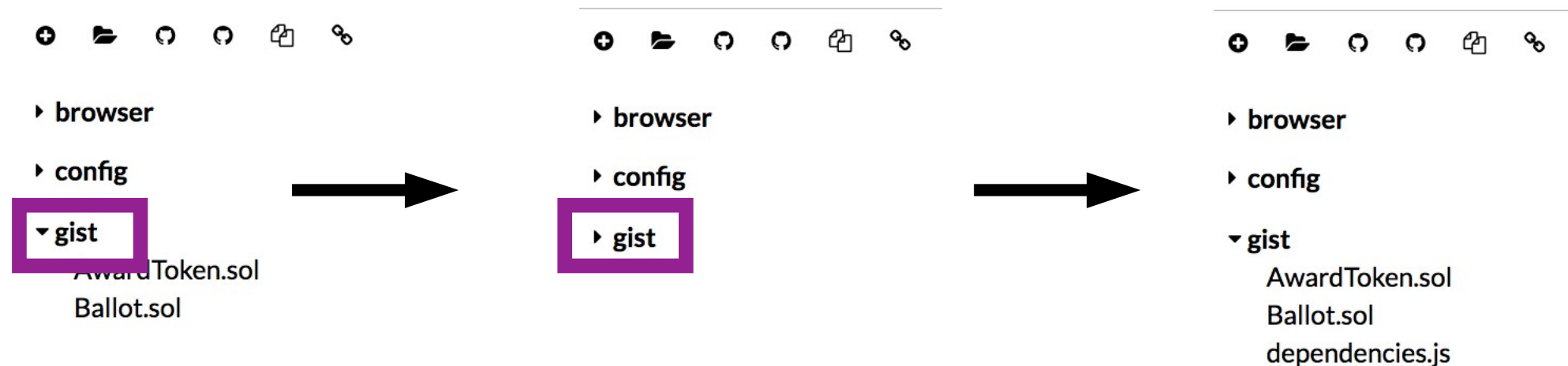
```
function AwardToken () {  
  ^-----^
```

Load from GITHUB

In the console type:

```
remix.loadgist('1d3250ca532b5f1e3ed4f52b18feb602')
```

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



Load up the dependencies

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



```
1 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-solidity/contracts/token/ERC20/ERC20Mintable.sol')
2 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-solidity/contracts/token/ERC20/IERC20.sol')
3 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-solidity/contracts/token/ERC20/ERC20.sol')
4 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-solidity/contracts/math/SafeMath.sol')
5 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-solidity/contracts/access/Roles.sol')
6 remix.loadurl('https://github.com/OpenZeppelin/openzeppelin-solidity/contracts/access/roles/MinterRole.sol')
7
```

2. In the console type: `remix.executeCurrent()`

-or-

copy each line of the dependencies.js into the console to load each file of the github repo

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

Update the code to point to the github repos

1. Go to AwardToken and update line 3

Replace:

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

With:

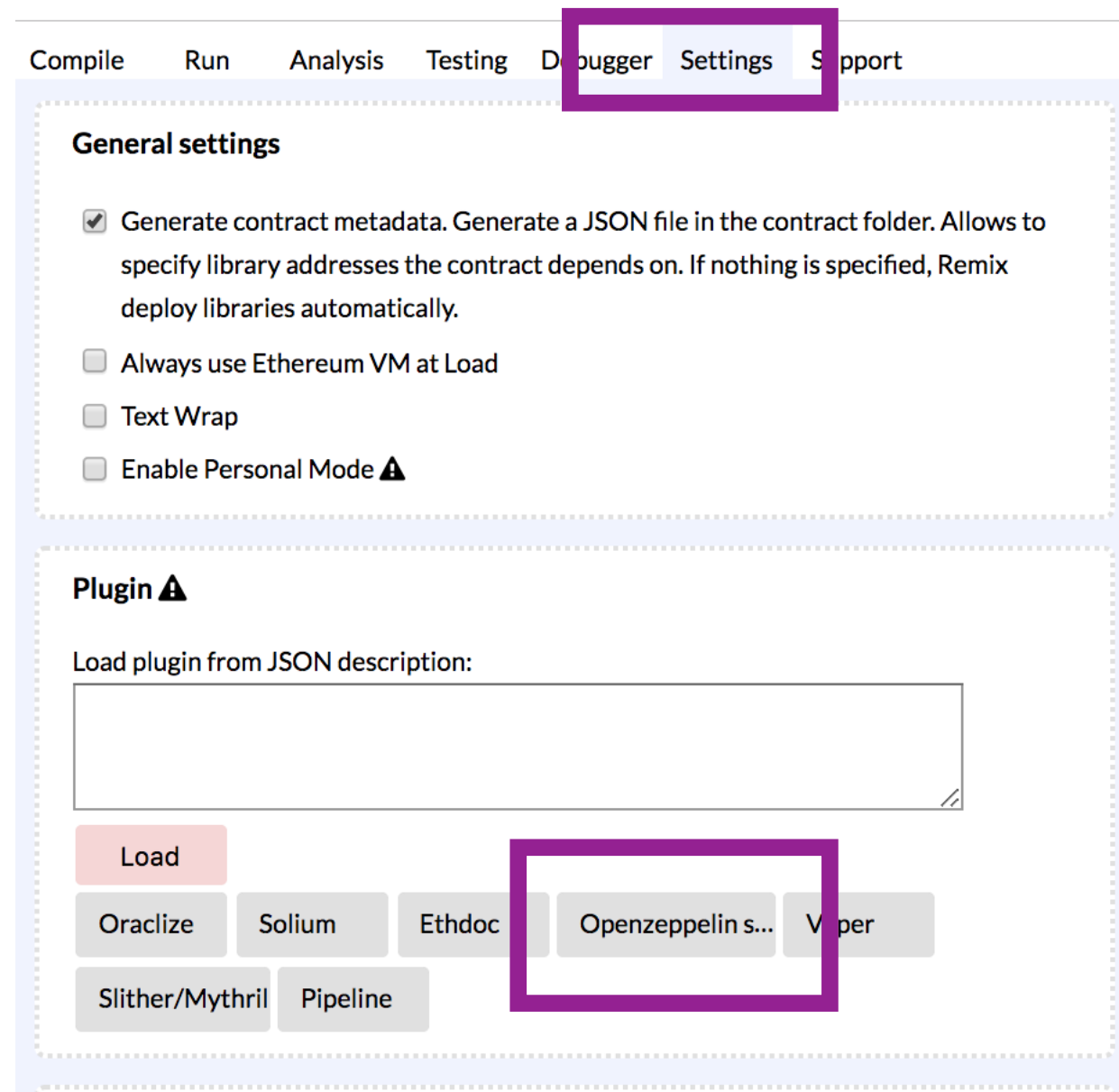
```
import "github/OpenZeppelin/openzeppelin-solidity/contracts/token/ERC20/ERC20Mintable.sol";
```

Now you are pointing to the downloaded repo in remix.

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

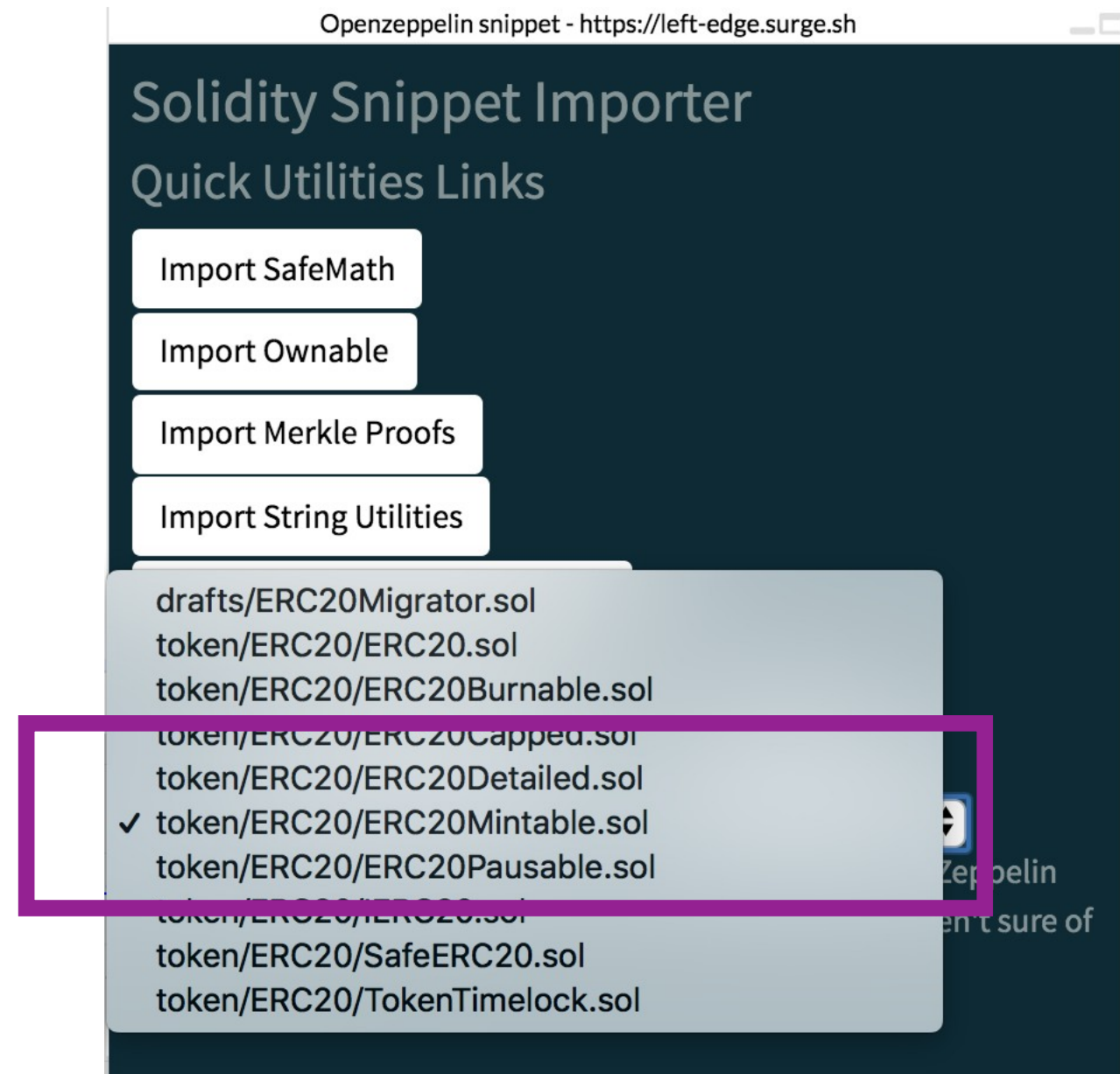
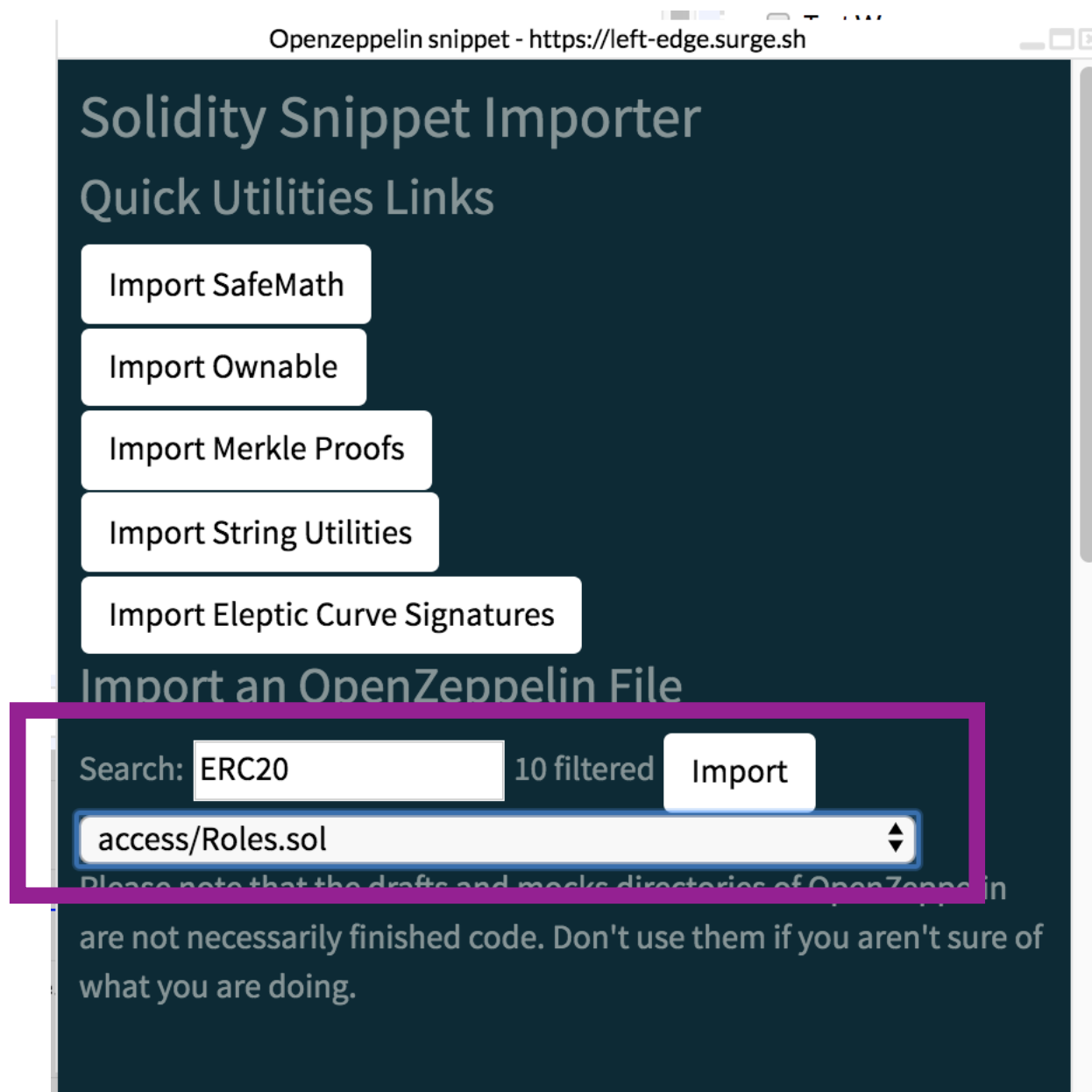
Load files from a url – via the Zeppelin plugin

Click the Openzeppelin plugin button in the settings tab



Choose ERC20Mintable.sol

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



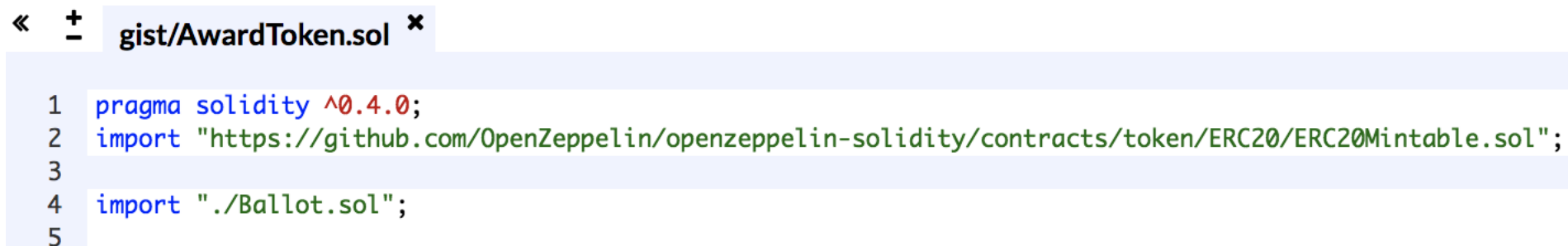
Update AwardToken.sol

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

So, delete this line:

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

AwardToken's import lines should look like this:



```
< ± gist/AwardToken.sol ×
1  pragma solidity ^0.4.0;
2  import "https://github.com/OpenZeppelin/openzeppelin-solidity/contracts/token/ERC20/ERC20Mintable.sol";
3
4  import './Ballot.sol';
5
```

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

Thank you

@ninabreznik @ryestew @yann300
@serapath @LianaHus @iurimatias
@GrandSchtroumpf

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