Lab 8: Bitcoin and Blockchain

1 Details

Aim: To provide a foundation in understanding in Bitcoin and Blockchain.

2 Activities

L1.1 Using blockchain.info, find the details of the genesis block:

Date created:

Reward:

Number of transactions:

Size of block:

Which account received the mining reward for the genesis block (last four digits):

How many USD does the original miner have in the account they used for the first genesis record:

When did the genesis block creator stop trading?

L1.2 Using blockchain.info, determine the following

Total bitcoins in circulation:

Most recent hash block (last four hex digits):

Block reward per block:

Difficulty:

Average time between blocks:

Market capitalisation (USD):

24 hr price (USD):

24hr transactions (USD):

Hash rate:

Last successful miner:

Maximum block size:

Balance for 1GbVUSW5WJmRCpaCJ4hanUny77oDaWW4to:

L1.3 Download and create the Python file defined on this page:

https://asecuritysite.com/encryption/bit

Now run the Python file, and compare the results in L.1.2.

Total bitcoins in circulation:

Most recent hash block (last four hex digits):

Block reward per block:

Difficulty:

Average time between blocks:

Market capitalisation (USD):

24 hr price (USD):

24hr transactions (USD):

Hash rate:

Balance for 1GbVUSW5WJmRCpaCJ4hanUny77oDaWW4to:

C Setting up your Ethereum wallet on Ropsten

The Ropsten network allows a user to test an Ethereum application, and using free Ether. Initially setup your MetaMask wallet. A document to outline how you set this up is here. Once you have set it up, answer the following:

- What is your public ID (just define the first four hex values)?
- Find out someone else's public ID, and send them 0.001 Ether. If you are doing the lab on your own, send it to Bill (ID: 0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233).
- Can you see the transaction on the Ethereum network? An example of a wallet is here.

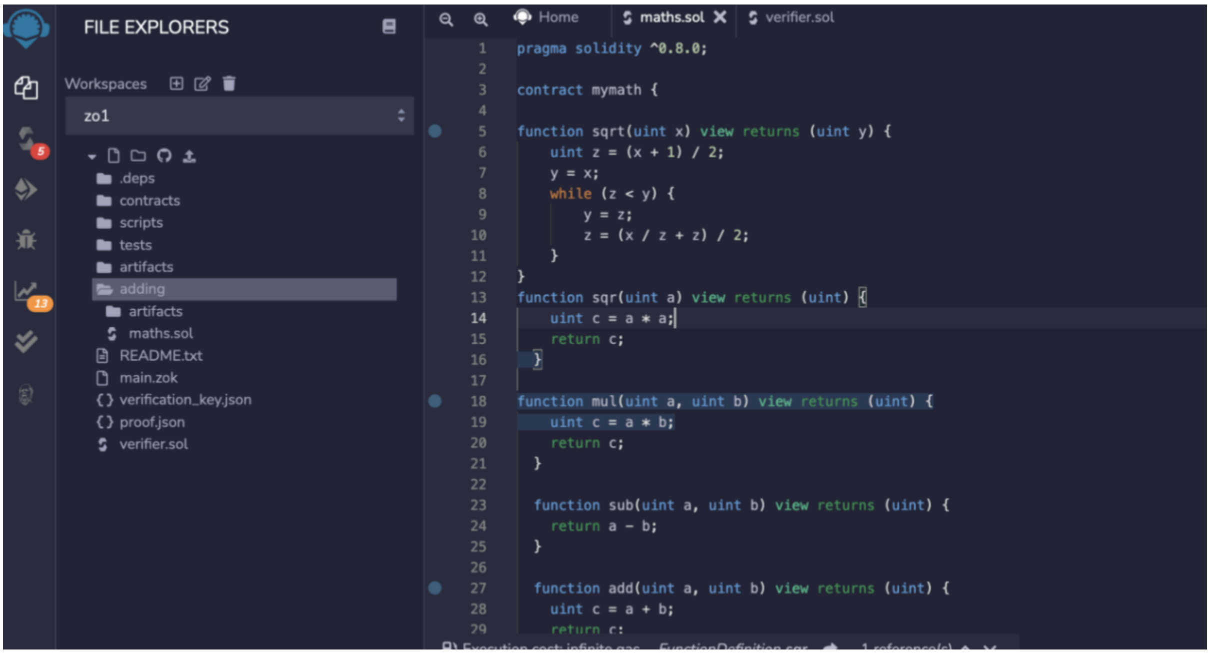
- Can you see your transaction on the Ethereum network for the person you send it to?
- What was the transaction fee for the transfer? If you were using the main Ethereum network, how much would the transaction cost in Dollars?
- Ask someone to send you 0.001 Ether. Did you receive it? If you are doing the lab on your own, ask your lab tutor to send you 0.001 Ether.

Creating a Smart Contract in Ethereum

So, let's write a bit of code that does some simple maths. In the following we will implement sqrt(), sqr(), mul(), sub(), and add():

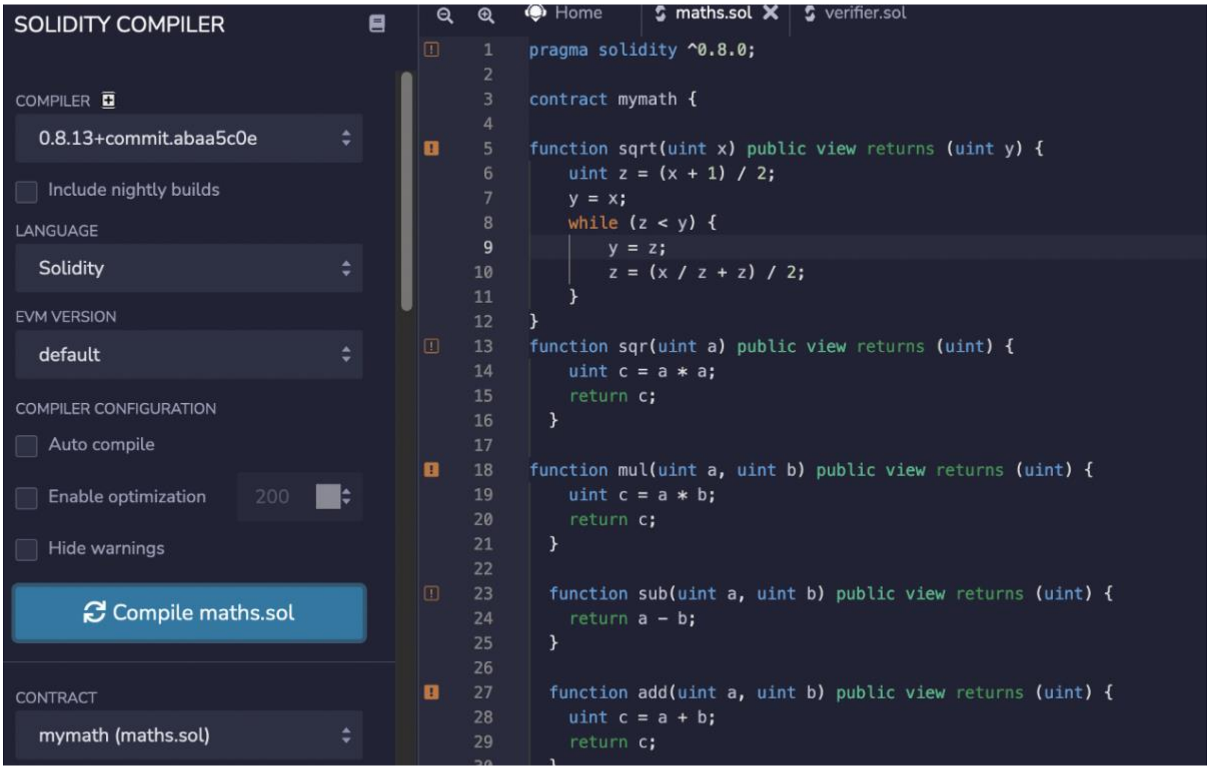
```
pragma solidity ^0.8.0;
contract mymath {function sqrt(uint x) public view returns (uint y) {
    uint z = (x + 1) / 2;
    y = x;
    while (z < y) {
        y = z;
        z = (x / z + z) / 2;
    }
}
function sqr(uint a) public view returns (uint) {
    uint c = a * a;
    return c;
  }
function mul(uint a, uint b) public view returns (uint) {
    uint c = a * b;
    return c;
  }
function sub(uint a, uint b) public view returns (uint) {
    return a - b;
  }
function add(uint a, uint b) public view returns (uint) {
    uint c = a + b;
    return c;
}}
```

In this case, the "public" part makes sure we can see the output of the function, and the "view" part allows it to be stateless (and where we just have to receiver the value without the smart contact remember the state). On Ethereum we normally use the Solidity language to create a smart contract and then compile it into the byte code required for the ledger. First, can we start by entering the Solidity code into Remix [here]:



```
pragma solidity ^0.8.0;

contract mymath {

function sqrt(uint x) view returns (uint y) {
    uint z = (x + 1) / 2;
    y = x;
    while (z < y) {
        y = z;
        z = (x / z + z) / 2;
    }
}
function sqr(uint a) view returns (uint) {
    uint c = a * a;
    return c;
  }

function mul(uint a, uint b) view returns (uint) {
    uint c = a * b;
    return c;
  }

  function sub(uint a, uint b) view returns (uint) {
    return a - b;
  }

  function add(uint a, uint b) view returns (uint) {
    uint c = a + b;
    return c;
```

Once entered, we can then compile it with the Solidity compiler. It is important to take a note of the compiler version at this stage, as we will need this later:



COMPILER: 0.8.13+commit.abaa5c0e

□ Include nightly builds

LANGUAGE: Solidity

EVM VERSION: default

COMPILER CONFIGURATION

□ Auto compile

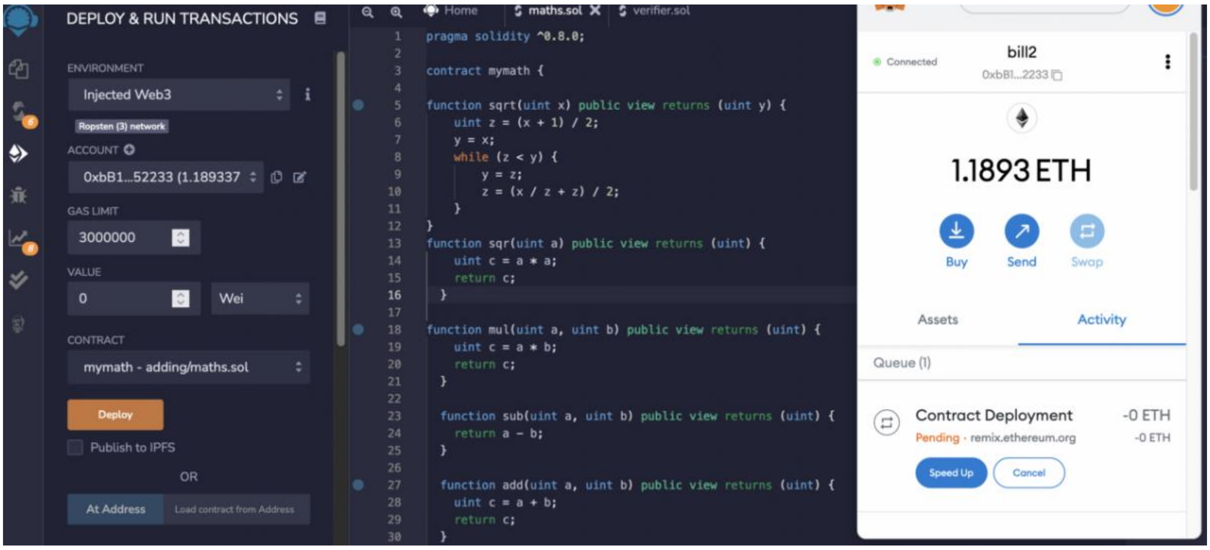
□ Enable optimization 200

□ Hide warnings

CONTRACT: mymath (maths.sol)

```
pragma solidity ^0.8.0;

contract mymath {

function sqrt(uint x) public view returns (uint y) {
    uint z = (x + 1) / 2;
    y = x;
    while (z < y) {
        y = z;
        z = (x / z + z) / 2;
    }
}
function sqr(uint a) public view returns (uint) {
    uint c = a * a;
    return c;
  }

function mul(uint a, uint b) public view returns (uint) {
    uint c = a * b;
    return c;
  }

  function sub(uint a, uint b) public view returns (uint) {
    return a - b;
  }

  function add(uint a, uint b) public view returns (uint) {
    uint c = a + b;
    return c;
```

Once compiled we can then deploy the smart contract to a test network (Ropsten). For this , we need to connect our Metamask wallet:



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Web3

Ropsten (3) network

ACCOUNT: 0xbB1...52233 (1.189337

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT: mymath - adding/maths.sol

```
pragma solidity ^0.8.0;

contract mymath {

function sqrt(uint x) public view returns (uint y) {
    uint z = (x + 1) / 2;
    y = x;
    while (z < y) {
        y = z;
        z = (x / z + z) / 2;
    }
}
function sqr(uint a) public view returns (uint) {
    uint c = a * a;
    return c;
  }

function mul(uint a, uint b) public view returns (uint) {
    uint c = a * b;
    return c;
  }

  function sub(uint a, uint b) public view returns (uint) {
    return a - b;
  }

  function add(uint a, uint b) public view returns (uint) {
    uint c = a + b;
    return c;
  }
```

Connected

bill2

0xbB1...2233

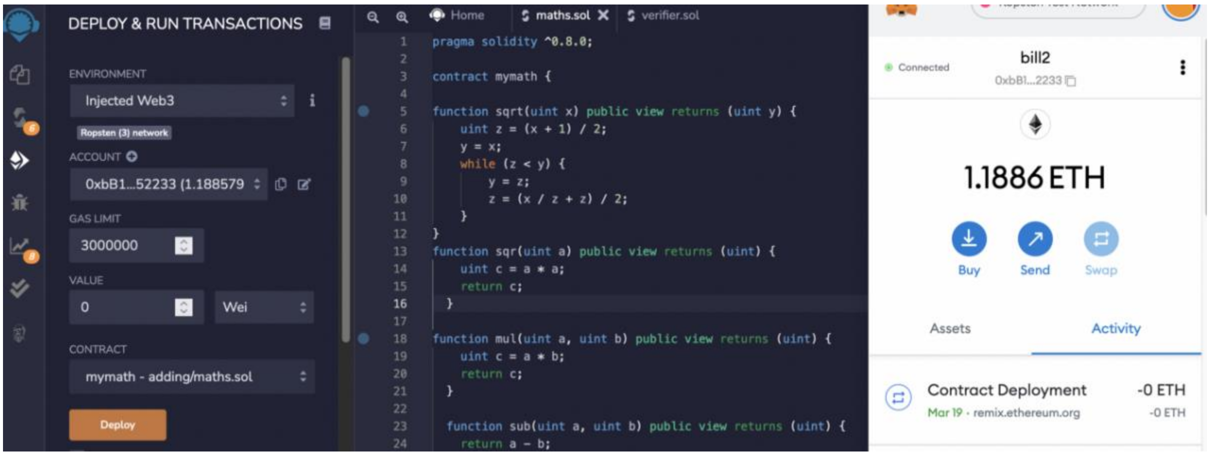
1.1893 ETH

Queue (1)

Contract Deployment -0 ETH

Pending · remix.ethereum.org -0 ETH

Once it has been deployed, we can see our wallet identifies the deployed contract:



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Web3

Ropsten (3) network

ACCOUNT: 0xbB1...52233 (1.188579

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT: mymath - adding/maths.sol

```
pragma solidity ^0.8.0;

contract mymath {

function sqrt(uint x) public view returns (uint y) {
    uint z = (x + 1) / 2;
    y = x;
    while (z < y) {
        y = z;
        z = (x / z + z) / 2;
    }
}
function sqr(uint a) public view returns (uint) {
    uint c = a * a;
    return c;
  }

function mul(uint a, uint b) public view returns (uint) {
    uint c = a * b;
    return c;
  }

  function sub(uint a, uint b) public view returns (uint) {
    return a - b;
```

Connected

bill2

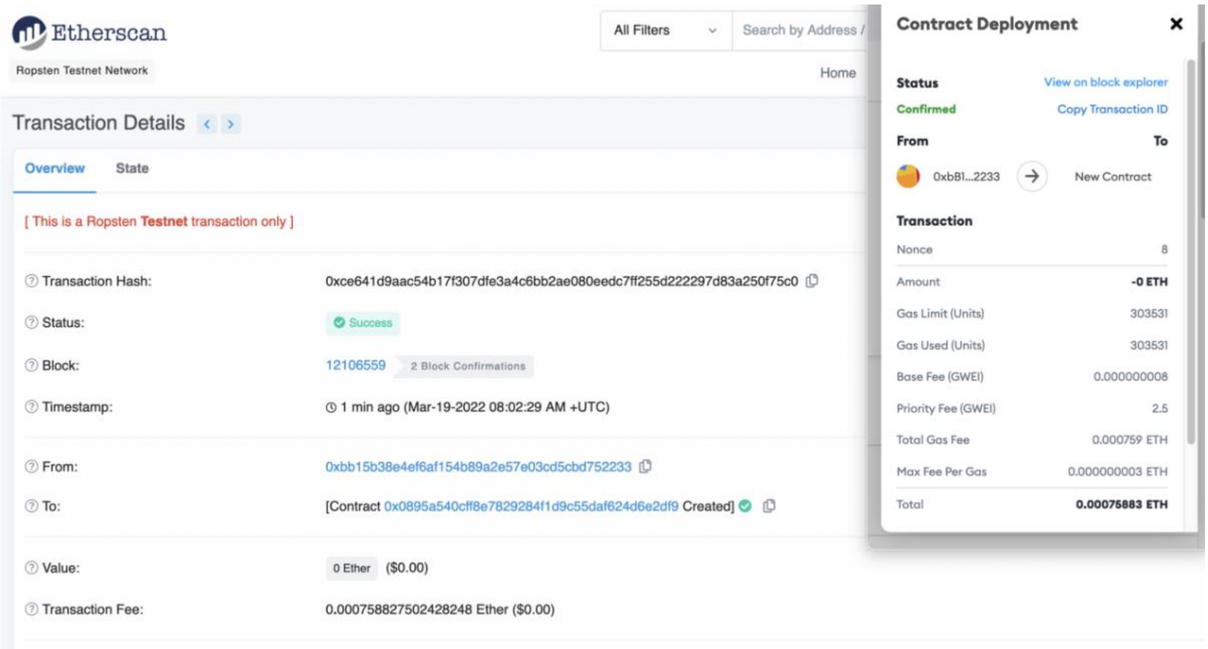
0xbB1...2233

1.1886 ETH

Contract Deployment -0 ETH

Mar 19 · remix.ethereum.org -0 ETH

And clicking through gives us the address of the contract, and then viewing it on the explorer, we can see the transaction:



Transaction Details

Overview State

[This is a Ropsten **Testnet** transaction only]

Transaction Hash: 0xce641d9aac54b17f307dfe3a4c6bb2ae080eedc7ff255d222297d83a250f75c0

Status: ✓ Success

Block: 12106559 2 Block Confirmations

Timestamp: 1 min ago (Mar-19-2022 08:02:29 AM +UTC)

From: 0xbb15b38e4ef6af154b89a2e57e03cd5cbd752233

To: [Contract 0x0895a540cff8e7829284f1d9c55daf624d6e2df9 Created] ✓

Value: 0 Ether ($0.00)

Transaction Fee: 0.000758827502428248 Ether ($0.00)

Contract Deployment ✕

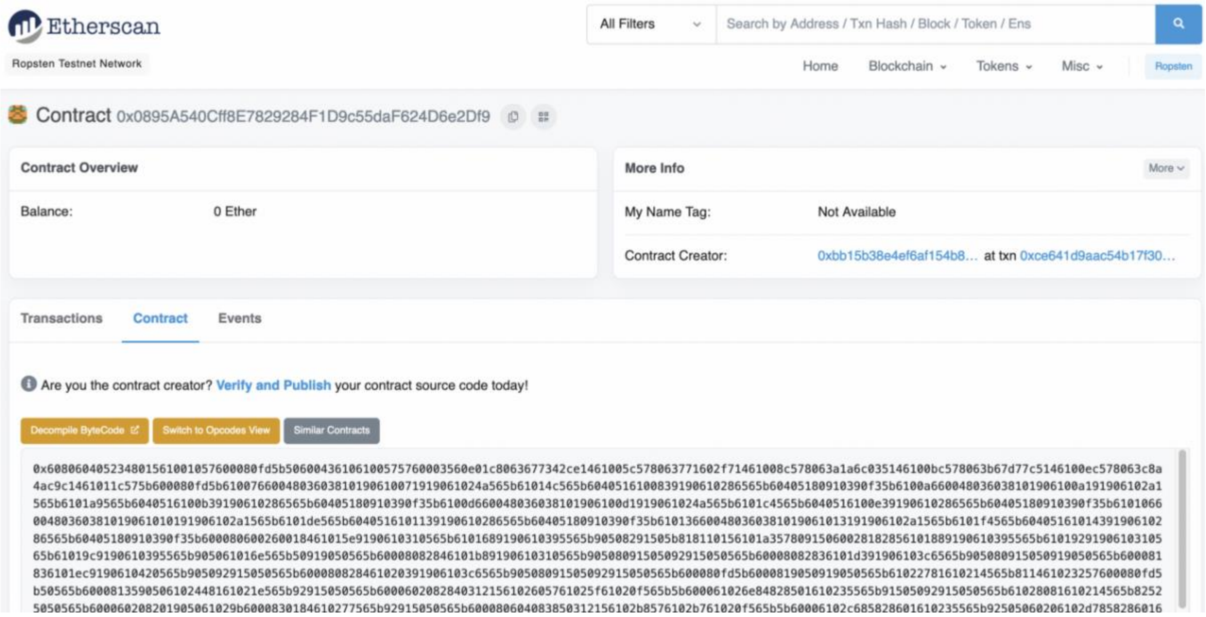
Status View on block explorer
Confirmed Copy Transaction ID

From **To**
0xbB1...2233 → New Contract

Transaction

Nonce	8
Amount	**-0 ETH**
Gas Limit (Units)	303531
Gas Used (Units)	303531
Base Fee (GWEI)	0.000000008
Priority Fee (GWEI)	2.5
Total Gas Fee	0.000759 ETH
Max Fee Per Gas	0.000000003 ETH
Total	**0.00075883 ETH**

The address here is “0x0895..”, so we can view the smart contract from: here. We now need to verify and publish the contact, with click on “Verify and Publish”:



Contract 0x0895A540Cff8E7829284F1D9c55daF624D6e2Df9

Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

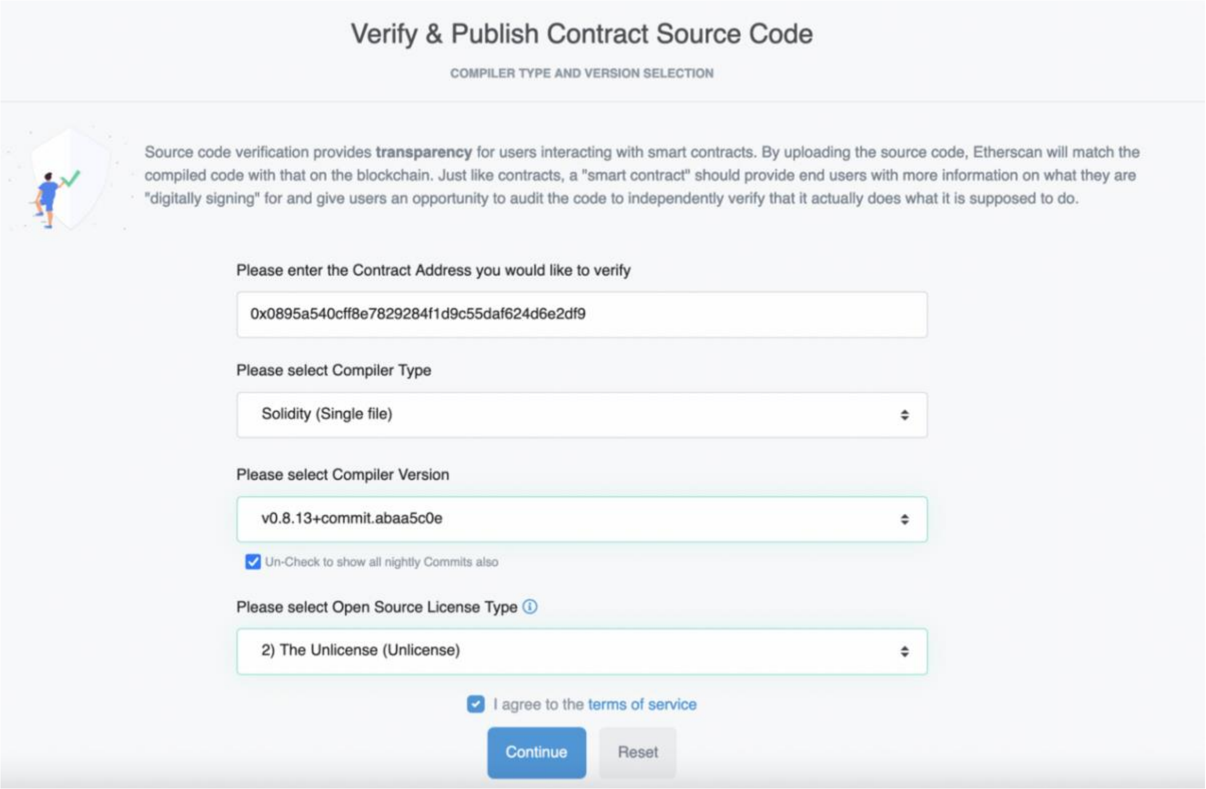
Contract Creator: 0xbb15b38e4ef6af154b8... at txn 0xce641d9aac54b17f30...

Transactions Contract Events

Are you the contract creator? **Verify and Publish** your contract source code today!

```
0x608060405234801561001057600080fd5b50600436106100575760003560e01c8063677342ce1461005c578063771602f71461008c578063a1a6c035146100bc578063b67d77c5146100ec578063c8a
4ac9c1461011c575b600080fd5b610076600480360381019061007191906102 4a565b61014c565b6040516100839190610286565b60405180910390f35b6100a660048036038101906100a191906102a1
565b6101a9565b6040516100b39190610286565b60405180910390f35b6100d660048036038101906100d1919061024a565b6101c4565b6040516100e39190610286565b60405180910390f35b6101066
00480360381019061010191906102a1565b6101de565b6040516101139190610286565b60405180910390f35b61013660048036038101906101319190610 2a1565b6101f4565b60405161014391906102
86565b60405180910390f35b600080600260018461015e9190610310565b6101689190610395565b905082915050 5b818110156101a357809150600281828561018891906103 95565b61019291906103105
65b61019c9190610395565b905061016e565b5091905056 5b60008082846101b89190610310565b905080915050929150505 65b60008082836101d391906103c6565b90508091505091905056 5b600081
836101ec9190610420565b905092915050565b600080828461020391906103c6565b905080915050929150505 65b600080fd5b6000819050919050565b610227816102145 65b811461023257600080fd5
b50565b6000813590506102448161021e565b92915050565b60006020828403121561026057610 25f61020f565b5b600061026e84828501610235565b91505092915050565b61028081610214565b8252
5050565b600060208201905061029b6000830184610277565b92915050565b6000806040838503121561 02b8576102b761020f565b5b60006102c685828601610235565b92505060206102d7858286016
```

After this, we can define the Compiler Version and the licence



Verify & Publish Contract Source Code

COMPILER TYPE AND VERSION SELECTION

Source code verification provides **transparency** for users interacting with smart contracts. By uploading the source code, Etherscan will match the compiled code with that on the blockchain. Just like contracts, a "smart contract" should provide end users with more information on what they are "digitally signing" for and give users an opportunity to audit the code to independently verify that it actually does what it is supposed to do.

Please enter the Contract Address you would like to verify: 0x0895a540cff8e7829284f1d9c55daf624d6e2df9

Please select Compiler Type: Solidity (Single file)

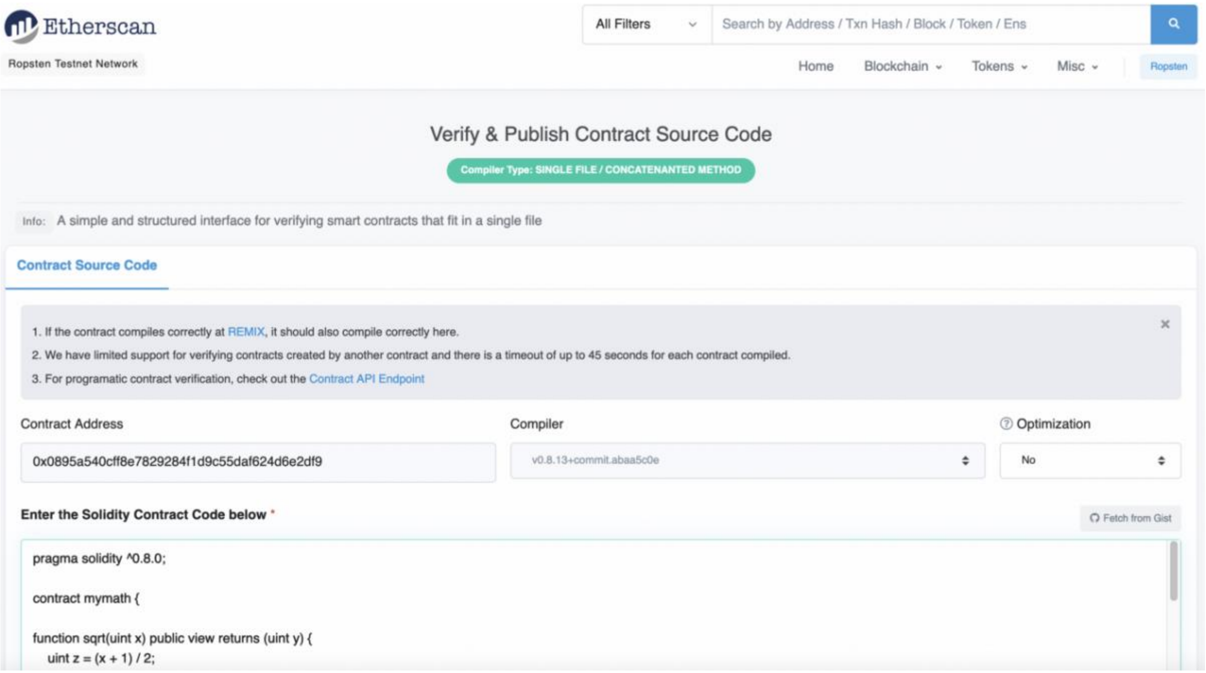
Please select Compiler Version: v0.8.13+commit.abaa5c0e

☑ Un-Check to show all nightly Commits also

Please select Open Source License Type: 2) The Unlicense (Unlicense)

☑ I agree to the terms of service

We then need to add your code for it to be checked:



Verify & Publish Contract Source Code

Compiler Type: SINGLE FILE / CONCATENANTED METHOD

Info: A simple and structured interface for verifying smart contracts that fit in a single file

Contract Source Code

1. If the contract compiles correctly at REMIX, it should also compile correctly here.
2. We have limited support for verifying contracts created by another contract and there is a timeout of up to 45 seconds for each contract compiled.
3. For programatic contract verification, check out the Contract API Endpoint

Contract Address: 0x0895a540cff8e7829284f1d9c55daf624d6e2df9

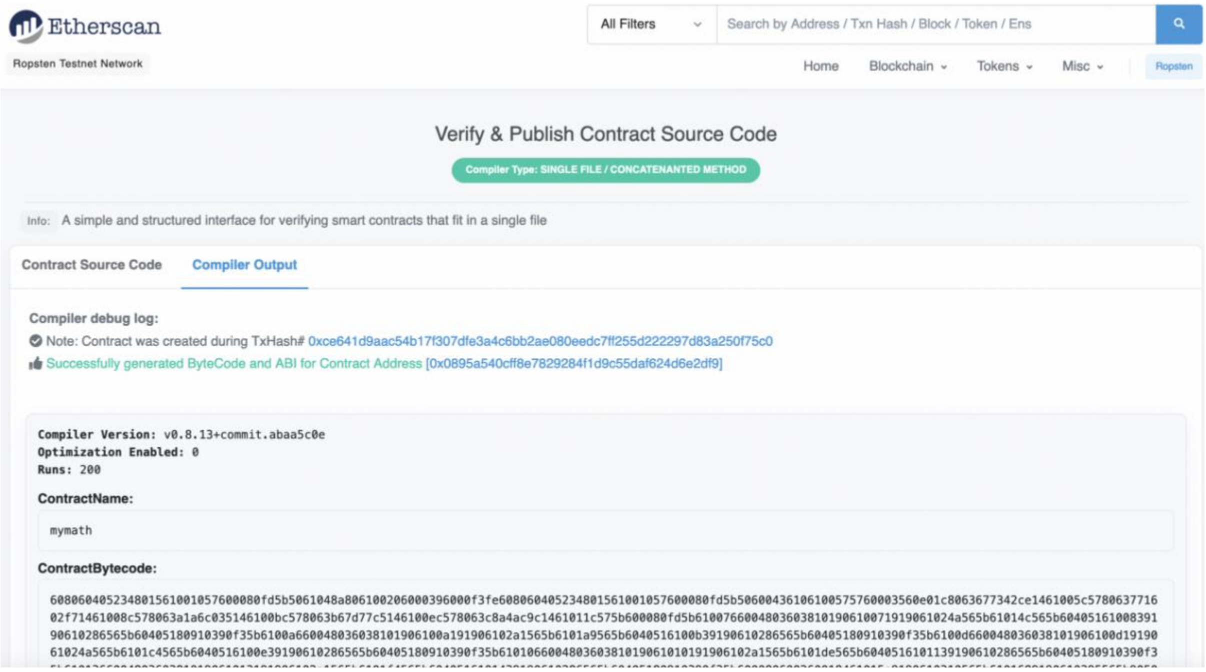
Compiler: v0.8.13+commit.abaa5c0e

Optimization: No

Enter the Solidity Contract Code below *

```
pragma solidity ^0.8.0;

contract mymath {

function sqrt(uint x) public view returns (uint y) {
   uint z = (x + 1) / 2;
```

It takes around 30 seconds, but, eventually, we should have our code accepted:



Verify & Publish Contract Source Code

Compiler Type: SINGLE FILE / CONCATENANTED METHOD

Info: A simple and structured interface for verifying smart contracts that fit in a single file

Compiler debug log:

Note: Contract was created during TxHash# 0xce641d9aac54b17f307dfe3a4c6bb2ae080eedc7ff255d222297d83a250f75c0

Successfully generated ByteCode and ABI for Contract Address [0x0895a540cff8e7829284f1d9c55daf624d6e2df9]

```
Compiler Version: v0.8.13+commit.abaa5c0e
Optimization Enabled: 0
Runs: 200
```

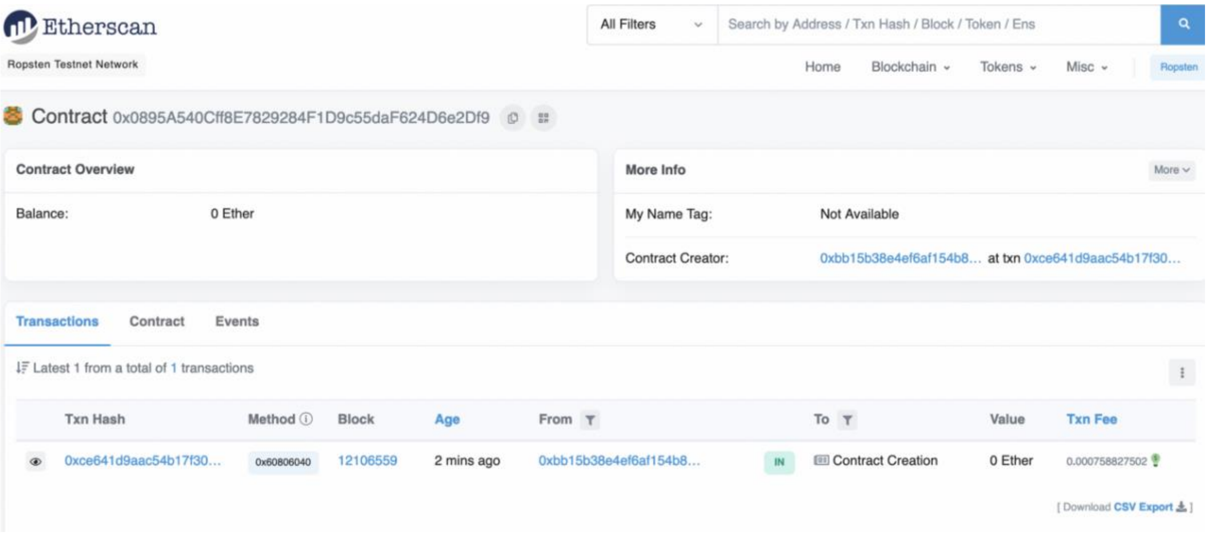
ContractName:

```
mymath
```

ContractBytecode:

```
608060405234801561001057600080fd5b5061048a806100206000396000f3fe608060405234801561001057600080fd5b50600436106100575760003560e01c8063677342ce1461005c5780637716
02f71461008c578063a1a6c035146100bc578063b67d77c5146100ec578063c8a4ac9c1461011c575b600080fd5b610076600480360381019061007191906102 4a565b61014c565b60405161008391
90610286565b60405180910390f35b6100a6600480360381019061 00a191906102a1565b6101a9565b6040516100b391906102 86565b60405180910390f35b6100d6600480360381019061 00d19190
61024a565b6101c4565b6040516100e391906102 86565b60405180910390f35b610106600480360381019061010191906102a1565b6101de565b60405161011391906102 86565b60405180910390f3
```

We now have the contract published to the Ropsten test network:



Contract 0x0895A540Cff8E7829284F1D9c55daF624D6e2Df9

Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

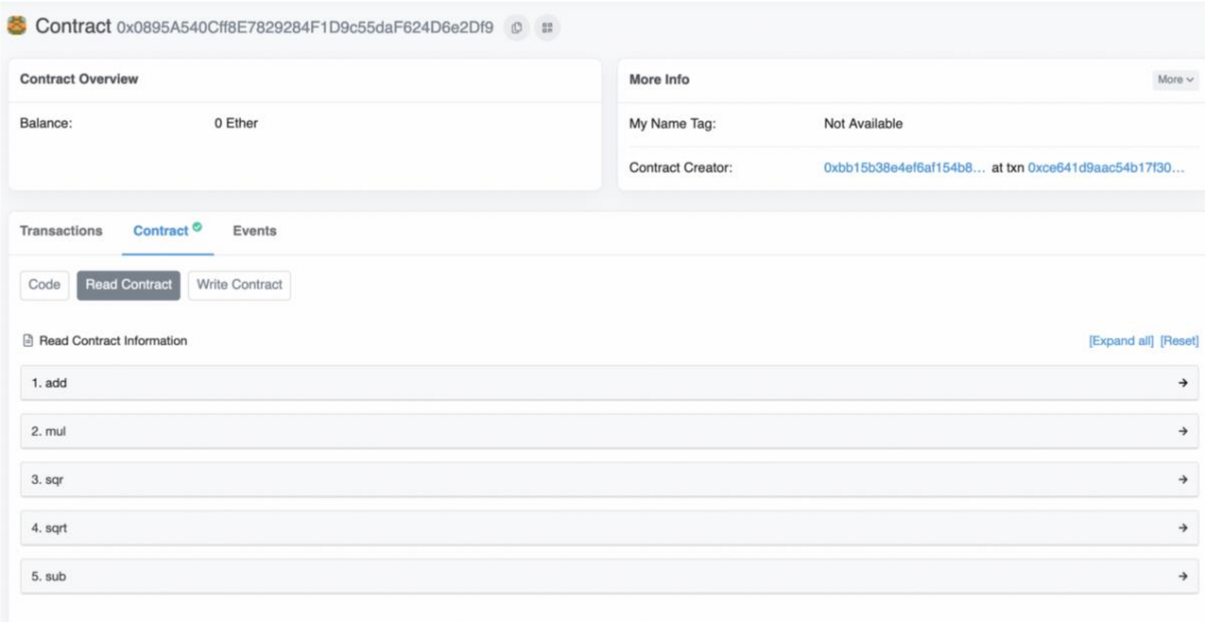
Contract Creator: 0xbb15b38e4ef6af154b8... at txn 0xce641d9aac54b17f30...

Latest 1 from a total of 1 transactions

	Txn Hash	Method	Block	Age	From		To	Value	Txn Fee
	0xce641d9aac54b17f30...	0x60806040	12106559	2 mins ago	0xbb15b38e4ef6af154b8...	IN	Contract Creation	0 Ether	0.000758827502

[Download CSV Export]

Next, by selected the Contract tab, and can view the read parameters. The exposed functions are add(), mul(), sqr(), sqrt() and sub():



Contract 0x0895A540Cff8E7829284F1D9c55daF624D6e2Df9

Contract Overview

Balance: 0 Ether

More Info

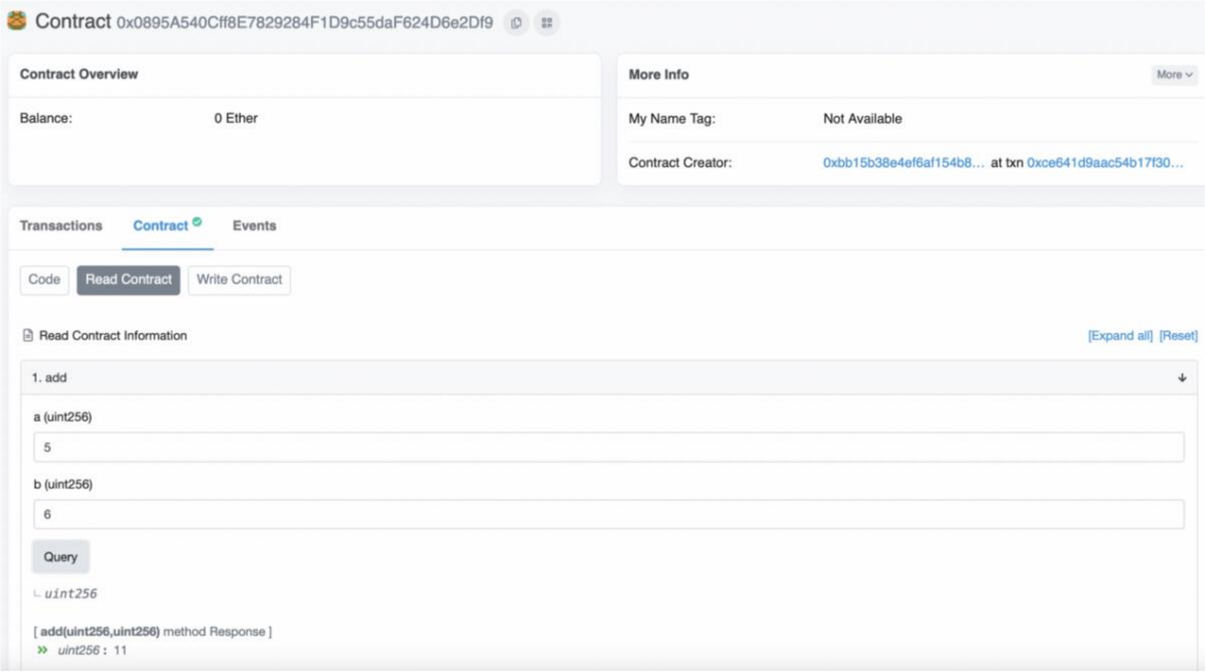
My Name Tag: Not Available

Contract Creator: 0xbb15b38e4ef6af154b8... at txn 0xce641d9aac54b17f30...

Read Contract Information

1. add
2. mul
3. sqr
4. sqrt
5. sub

To test, we can just enter the variables for a given function, and get a result:



Contract 0x0895A540Cff8E7829284F1D9c55daF624D6e2Df9

Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

Contract Creator: 0xbb15b38e4ef6af154b8... at txn 0xce641d9aac54b17f30...

Read Contract Information

1. add

a (uint256)

5

b (uint256)

6

└ uint256

[add(uint256,uint256) method Response]

» uint256 : 11

Creating ERC-20 tokens

Within the Ethereum blockchain, we can record transactions and run smart contracts. These things allow us to run DApps (decentralized applications) and which can support the running of the infrastructure in return for some payment (Ether). A DApp can also create tokens for new currencies, shares in a company or to prove the ownership of an asset. ERC-20 is a standard format for a Fungible Token and which can support the sharing, transfer and storage of tokens. These tokens are supported by the whole of the Ethereum infrastructure and can be easily traded. They support a number of mandatory functions:

- totalSupply. This function is the total number of ERC-20 tokens that have been created.
- balanceOf. This function identifies the number of tokens that a given address has in its account.
- transfer. This function supports the transfer of tokens to a defined user address.
- transferFrom. This function supports a user to transfer tokens to another user.
- approve. This function checks that a transaction is valid, based on the supply of token.
- allowance. This function checks if a user has enough funds in their account for a transaction.

There are also a number of options:

- Token Name. This is the name that the token will be defined as.
- Symbol. This is the symbol that the token will use.
- Decimal. This is the number of decimal places to be used for any transactions.

Now we you create your own token. If you are Bob Smith, then call your token "BobSmithToken", and your currency will be "BobSmith".

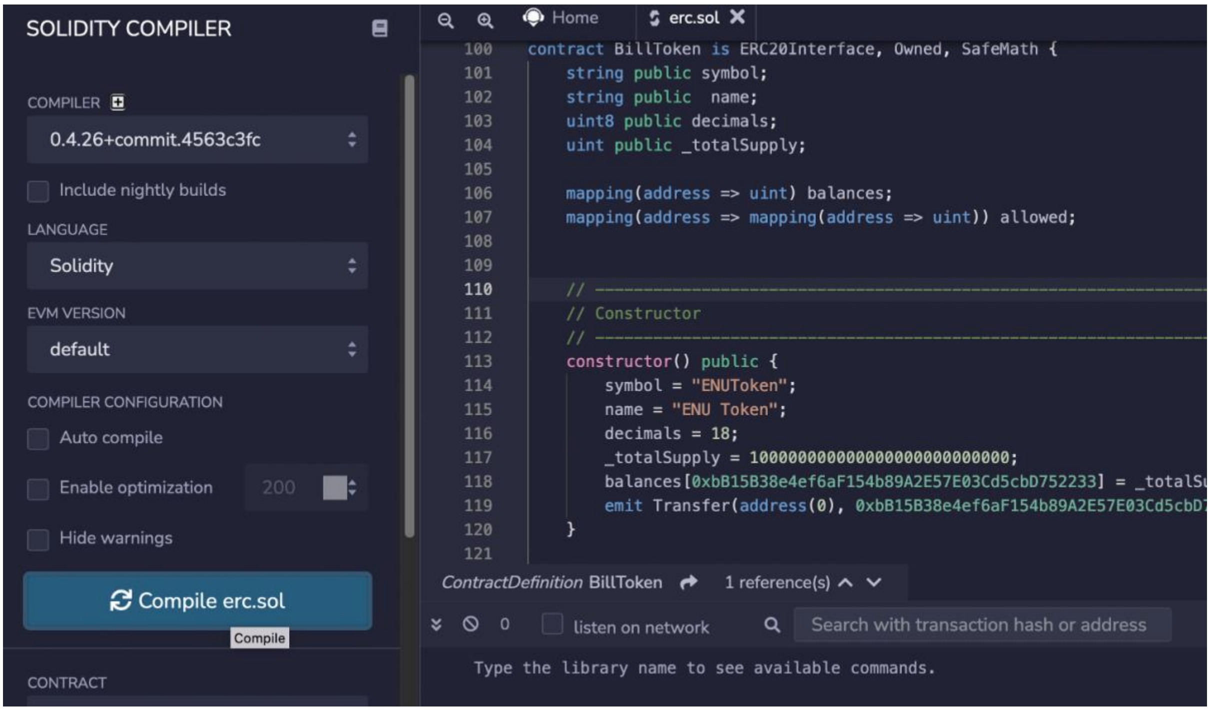
So, let's create a token named "ENUToken" (change the name to your name), and use the tutorial sample from here. First, we open up https://remix.ethereum.org/, and enter the following Solidy contract:

```
pragma solidity ^0.4.24;

// ----------------------------------------------------------------------------
// 'ENU Token' token contract
//
// Deployed to : 0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233
// Symbol      : ENUToken
// Name        : ENU Token
// Total supply: 100000000
// Decimals    : 18

// Based on https://github.com/bitfwdcommunity/Issue-your-own-ERC20-token/tree/master/contracts


// ----------------------------------------------------------------------------
// Safe maths
// ----------------------------------------------------------------------------
contract SafeMath {
    function safeAdd(uint a, uint b) public pure returns (uint c) {
        c = a + b;
        require(c >= a);
    }
    function safeSub(uint a, uint b) public pure returns (uint c) {
        require(b <= a);
        c = a - b;
    }
    function safeMul(uint a, uint b) public pure returns (uint c) {
        c = a * b;
        require(a == 0 || c / a == b);
    }
    function safeDiv(uint a, uint b) public pure returns (uint c) {
        require(b > 0);
        c = a / b;
    }
}


// ----------------------------------------------------------------------------
----
// ERC Token Standard #20 Interface
// https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20-token-
standard.md
// ----------------------------------------------------------------------------
----
contract ERC20Interface {
    function totalSupply() public constant returns (uint);
    function balanceOf(address tokenOwner) public constant returns (uint 
balance);
    function allowance(address tokenOwner, address spender) public constant 
returns (uint remaining);
    function transfer(address to, uint tokens) public returns (bool 
success);
    function approve(address spender, uint tokens) public returns (bool 
success);
    function transferFrom(address from, address to, uint tokens) public 
returns (bool success);

    event Transfer(address indexed from, address indexed to, uint tokens);
    event Approval(address indexed tokenOwner, address indexed spender, 
uint tokens);
}


// ----------------------------------------------------------------------------
----
// Contract function to receive approval and execute function in one call
//
// Borrowed from MiniMeToken
// ----------------------------------------------------------------------------
----
contract ApproveAndCallFallBack {
    function receiveApproval(address from, uint256 tokens, address token, 
bytes data) public;
}


// ----------------------------------------------------------------------------
----
// Owned contract
// ----------------------------------------------------------------------------
----
contract Owned {
    address public owner;
    address public newOwner;

    event OwnershipTransferred(address indexed _from, address indexed _to);

    constructor() public {
        owner = msg.sender;
    }

    modifier onlyOwner {
        require(msg.sender == owner);
        _;
    }

    function transferOwnership(address _newOwner) public onlyOwner {
        newOwner = _newOwner;
    }
    function acceptOwnership() public {
        require(msg.sender == newOwner);
        emit OwnershipTransferred(owner, newOwner);
        owner = newOwner;
        newOwner = address(0);
    }
}


// ------------------------------------------------------------------------
----
// ERC20 Token, with the addition of symbol, name and decimals and assisted
// token transfers
// ------------------------------------------------------------------------
----
contract <b style="color:red">BillToken</b> is ERC20Interface, Owned,
SafeMath {
    string public symbol;
    string public  name;
    uint8 public decimals;
    uint public _totalSupply;

    mapping(address => uint) balances;
    mapping(address => mapping(address => uint)) allowed;


    // --------------------------------------------------------------------
----
    // Constructor
    // --------------------------------------------------------------------
----
    constructor() public {
        symbol = "ENUToken";
        name = "ENU Token";
        decimals = 18;
        _totalSupply = 100000000000000000000000000;
        balances[0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233] =
_totalSupply;
        emit Transfer(address(0), <b
style="color:red">0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233</b>,
_totalSupply);
    }


    // --------------------------------------------------------------------
----
    // Total supply
    // --------------------------------------------------------------------
----
    function totalSupply() public constant returns (uint) {
        return _totalSupply  - balances[address(0)];
    }


    // --------------------------------------------------------------------
----
    // Get the token balance for account tokenOwner
    // --------------------------------------------------------------------
----
    function balanceOf(address tokenOwner) public constant returns (uint
balance) {
        return balances[tokenOwner];
    }


    // --------------------------------------------------------------------
----
    // Transfer the balance from token owner's account to to account
    // - Owner's account must have sufficient balance to transfer
    // - 0 value transfers are allowed
    // --------------------------------------------------------------------
----
    function transfer(address to, uint tokens) public returns (bool
success) {
        balances[msg.sender] = safeSub(balances[msg.sender], tokens);
        balances[to] = safeAdd(balances[to], tokens);
        emit Transfer(msg.sender, to, tokens);
        return true;
    }


    // --------------------------------------------------------------------
----
    // Token owner can approve for spender to transferFrom(...) tokens
    // from the token owner's account
    //
    // https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20-token-
standard.md
    // recommends that there are no checks for the approval double-spend
attack
    // as this should be implemented in user interfaces
    // --------------------------------------------------------------------
----
    function approve(address spender, uint tokens) public returns (bool
success) {
        allowed[msg.sender][spender] = tokens;
        emit Approval(msg.sender, spender, tokens);
        return true;
    }


    // --------------------------------------------------------------------
----
    // Transfer tokens from the from account to the to account
    //
    // The calling account must already have sufficient tokens
approve(...)-d
    // for spending from the from account and
    // - From account must have sufficient balance to transfer
    // - Spender must have sufficient allowance to transfer
    // - 0 value transfers are allowed
    // --------------------------------------------------------------------
----
    function transferFrom(address from, address to, uint tokens) public
returns (bool success) {
        balances[from] = safeSub(balances[from], tokens);
        allowed[from][msg.sender] = safeSub(allowed[from][msg.sender],
tokens);
        balances[to] = safeAdd(balances[to], tokens);
        emit Transfer(from, to, tokens);
        return true;
    }


    // --------------------------------------------------------------------
----
    // Returns the amount of tokens approved by the owner that can be
    // transferred to the spender's account
    // --------------------------------------------------------------------
----
    function allowance(address tokenOwner, address spender) public constant
returns (uint remaining) {
        return allowed[tokenOwner][spender];
    }


    // --------------------------------------------------------------------
----
    // Token owner can approve for spender to transferFrom(...) tokens
    // from the token owner's account. The spender contract function
    // receiveApproval(...) is then executed
    // --------------------------------------------------------------------
----
    function approveAndCall(address spender, uint tokens, bytes data)
public returns (bool success) {
        allowed[msg.sender][spender] = tokens;
        emit Approval(msg.sender, spender, tokens);
        ApproveAndCallFallBack(spender).receiveApproval(msg.sender, tokens,
this, data);
        return true;
    }


    // --------------------------------------------------------------------
----
    // Don't accept ETH
    // --------------------------------------------------------------------
----
    function () public payable {
        revert();
    }

    // --------------------------------------------------------------------
----
    // Owner can transfer out any accidentally sent ERC20 tokens
    // --------------------------------------------------------------------
----
    function transferAnyERC20Token(address tokenAddress, uint tokens)
public onlyOwner returns (bool success) {
        return ERC20Interface(tokenAddress).transfer(owner, tokens);
    }
}
```

When you create your own contract, make sure you change the public constructor() with: the **symbol**, the **name**, and the **wallet ID**. You are the owner of the token, so you need to enter the public ID of your wallet for two hex values given next:

```
    constructor() public {
    symbol = "ENUToken";
    name = "ENU Token";
    decimals = 18;
    _totalSupply = 100000000000000000000000000;
    balances[0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233] = _totalSupply;
    emit Transfer(address(0), 0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233,
_totalSupply);
    }
```

The wallet ID is the public ID of your wallet in MetaMask. Now we compile:



SOLIDITY COMPILER

COMPILER: 0.4.26+commit.4563c3fc

□ Include nightly builds

LANGUAGE: Solidity

EVM VERSION: default

COMPILER CONFIGURATION

□ Auto compile

□ Enable optimization 200

□ Hide warnings

Compile erc.sol

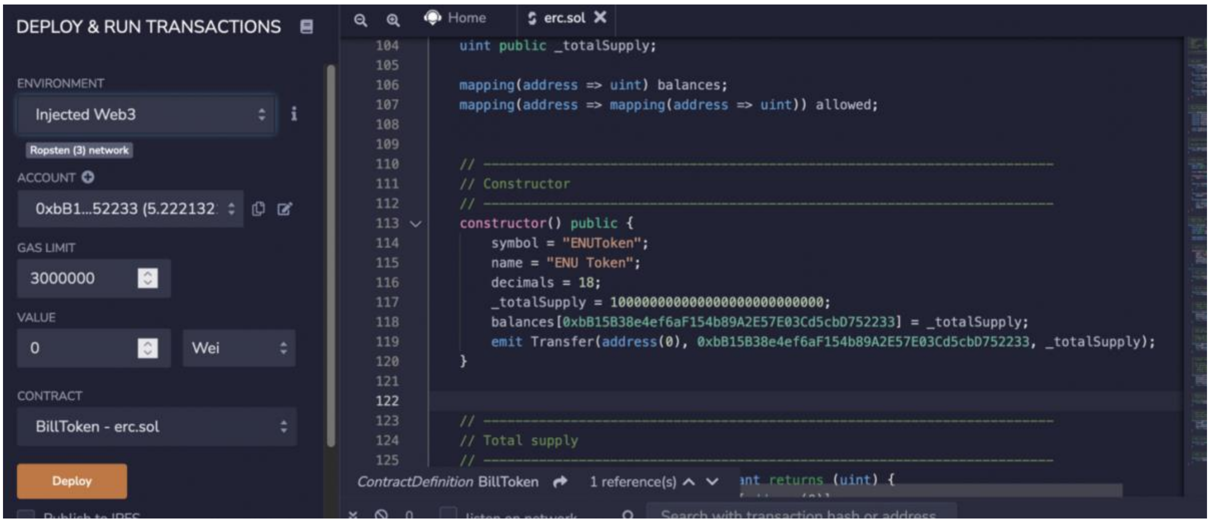
CONTRACT

```
contract BillToken is ERC20Interface, Owned, SafeMath {
    string public symbol;
    string public  name;
    uint8 public decimals;
    uint public _totalSupply;

    mapping(address => uint) balances;
    mapping(address => mapping(address => uint)) allowed;


    // ------------------------------------------------------------------------
    // Constructor
    // ------------------------------------------------------------------------
    constructor() public {
        symbol = "ENUToken";
        name = "ENU Token";
        decimals = 18;
        _totalSupply = 100000000000000000000000000;
        balances[0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233] = _totalSu
        emit Transfer(address(0), 0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD7
    }

```

ContractDefinition BillToken 1 reference(s)

```
Type the library name to see available commands.
```

Next, we will deploy to the Ropsten test network:



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Web3

Ropsten (3) network

ACCOUNT: 0xbB1...52233 (5.222132

GAS LIMIT: 3000000

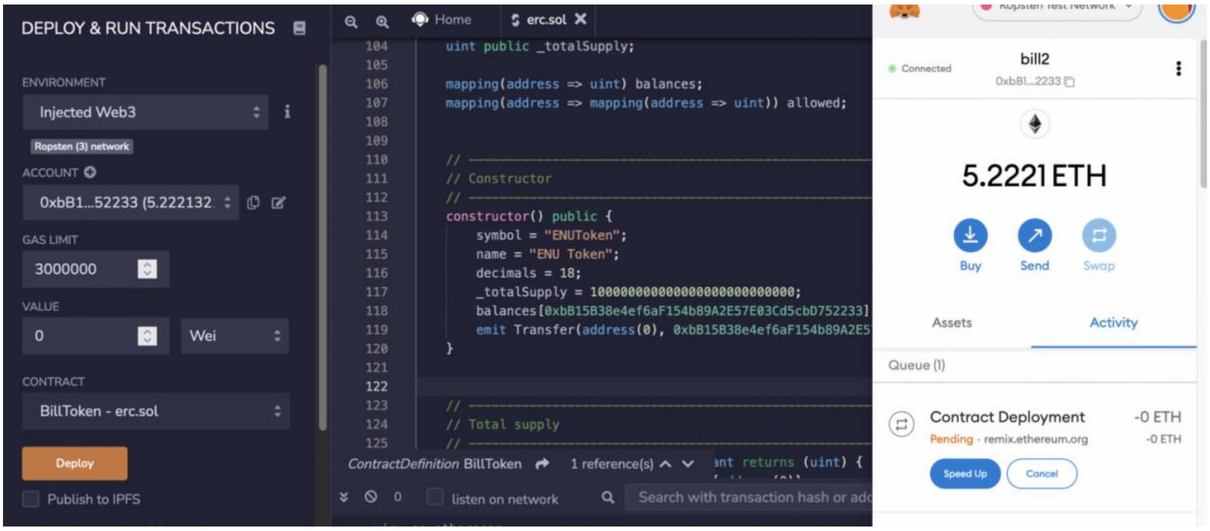
VALUE: 0 Wei

CONTRACT: BillToken - erc.sol

```
    uint public _totalSupply;

    mapping(address => uint) balances;
    mapping(address => mapping(address => uint)) allowed;


    // ------------------------------------------------------------------------
    // Constructor
    // ------------------------------------------------------------------------
    constructor() public {
        symbol = "ENUToken";
        name = "ENU Token";
        decimals = 18;
        _totalSupply = 100000000000000000000000000;
        balances[0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233] = _totalSupply;
        emit Transfer(address(0), 0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233, _totalSupply);
    }


    // ------------------------------------------------------------------------
    // Total supply
    // ------------------------------------------------------------------------
```

ContractDefinition BillToken 1 reference(s)

After this, our contract will be shown as being pending deployment:



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Web3

Ropsten (3) network

ACCOUNT: 0xbB1...52233 (5.222132

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT: BillToken - erc.sol

Deploy

□ Publish to IPFS

```
    uint public _totalSupply;

    mapping(address => uint) balances;
    mapping(address => mapping(address => uint)) allowed;


    // ------------------------------------------------------------------------
    // Constructor
    // ------------------------------------------------------------------------
    constructor() public {
        symbol = "ENUToken";
        name = "ENU Token";
        decimals = 18;
        _totalSupply = 1000000000000000000000000000;
        balances[0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233]
        emit Transfer(address(0), 0xbB15B38e4ef6aF154b89A2E5
    }


    // ------------------------------------------------------------------------
    // Total supply
    // ------------------------------------------------------------------------
```

ContractDefinition BillToken 1 reference(s)

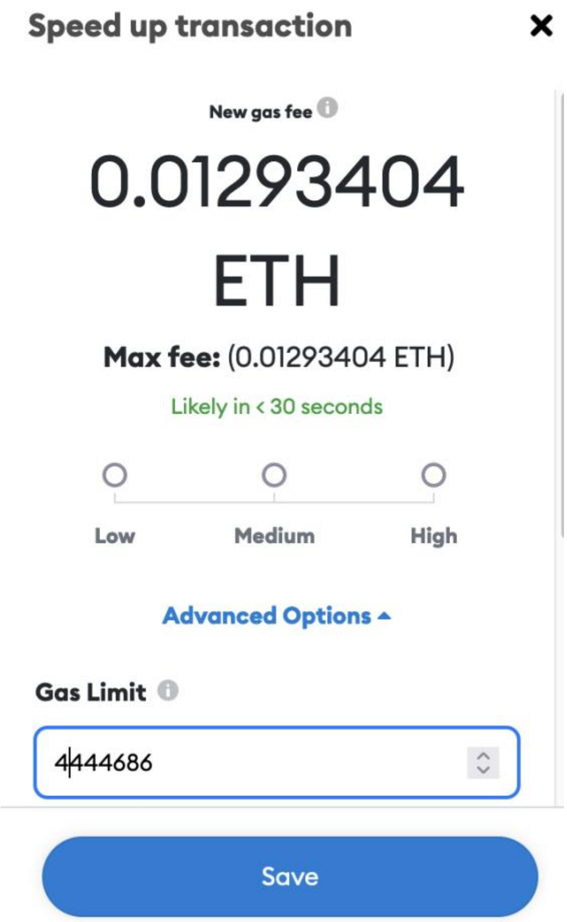
□ listen on network

Search with transaction hash or add

Contract Deployment -0 ETH

Pending · remix.ethereum.org -0 ETH

It will take 10–15 minutes to deploy, but it can be speeded up by increasing the gas limit:



Speed up transaction

New gas fee

0.01293404 ETH

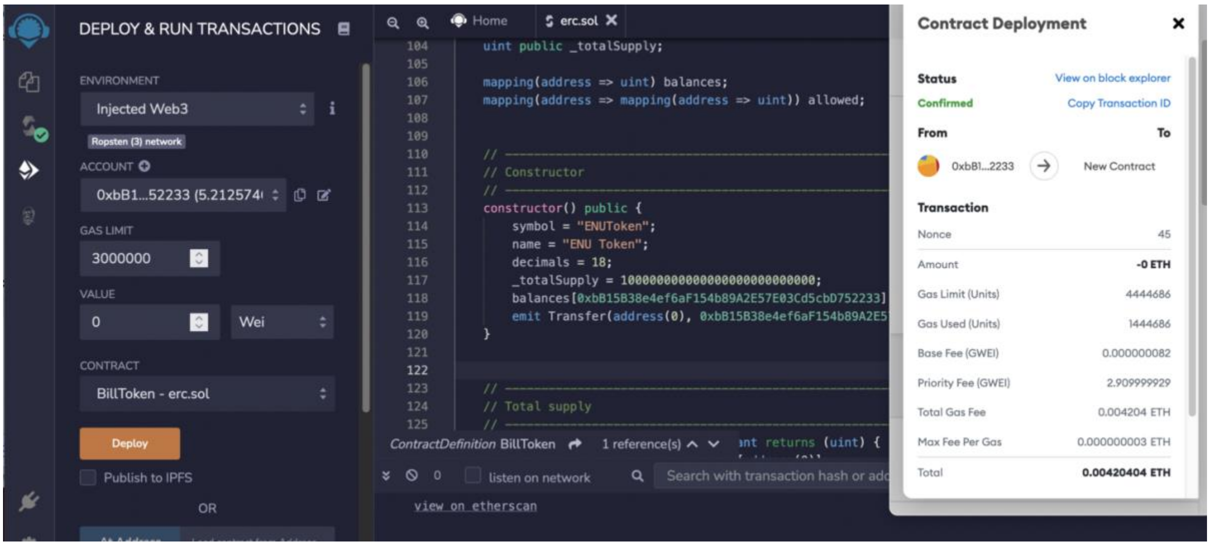
Max fee: (0.01293404 ETH)

Likely in < 30 seconds

○ Low ○ Medium ○ High

Gas Limit: 4444686

Once deployed, we can view the contract details:



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Web3

Ropsten (3) network

ACCOUNT: 0xbB1...52233 (5.212574

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT: BillToken - erc.sol

```
    uint public _totalSupply;

    mapping(address => uint) balances;
    mapping(address => mapping(address => uint)) allowed;


    // -----------------------------------------------------------
    // Constructor
    // -----------------------------------------------------------
    constructor() public {
        symbol = "ENUToken";
        name = "ENU Token";
        decimals = 18;
        _totalSupply = 100000000000000000000000000;
        balances[0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233]
        emit Transfer(address(0), 0xbB15B38e4ef6aF154b89A2E5
    }


    // -----------------------------------------------------------
    // Total supply
    // -----------------------------------------------------------
```

ContractDefinition BillToken 1 reference(s)

Contract Deployment

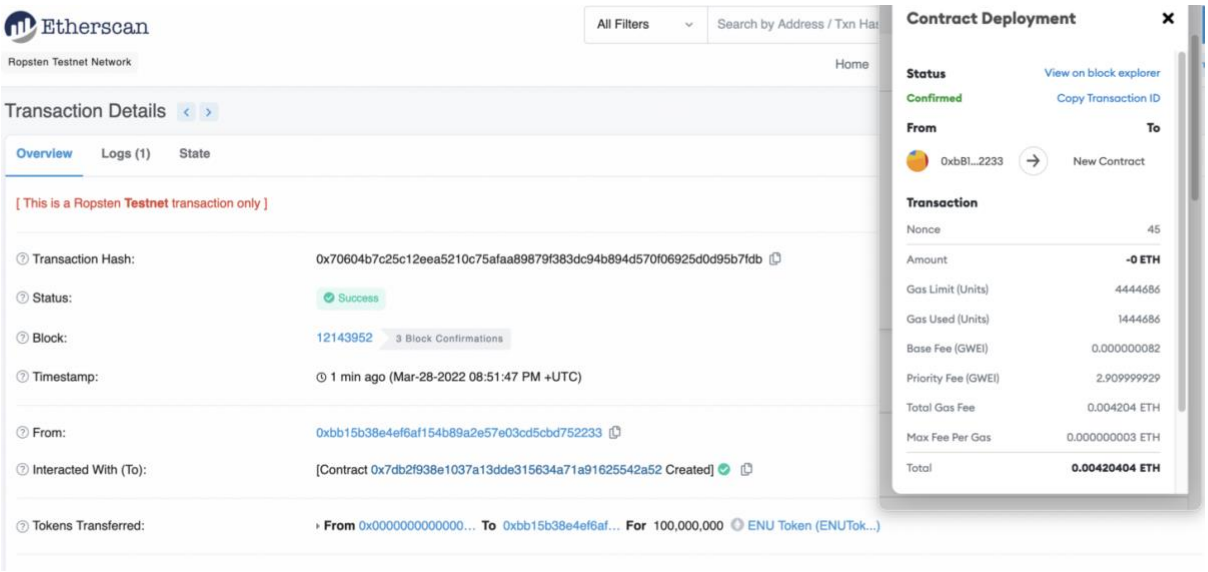
Status: Confirmed

From: 0xbB1...2233 → To: New Contract

Transaction

Nonce	45
Amount	-0 ETH
Gas Limit (Units)	4444686
Gas Used (Units)	1444686
Base Fee (GWEI)	0.000000082
Priority Fee (GWEI)	2.909999929
Total Gas Fee	0.004204 ETH
Max Fee Per Gas	0.000000003 ETH
Total	**0.00420404 ETH**

And can then view the transaction for the contact [here]:



Transaction Details

[This is a Ropsten **Testnet** transaction only]

Transaction Hash: 0x70604b7c25c12eea5210c75afaa89879f383dc94b894d570f06925d0d95b7fdb

Status: Success

Block: 12143952 3 Block Confirmations

Timestamp: 1 min ago (Mar-28-2022 08:51:47 PM +UTC)

From: 0xbb15b38e4ef6af154b89a2e57e03cd5cbd752233

Interacted With (To): [Contract 0x7db2f938e1037a13dde315634a71a91625542a52 Created]

Tokens Transferred: **From** 0x0000000000000... **To** 0xbb15b38e4ef6af... **For** 100,000,000 ENU Token (ENUTok...)

Contract Deployment

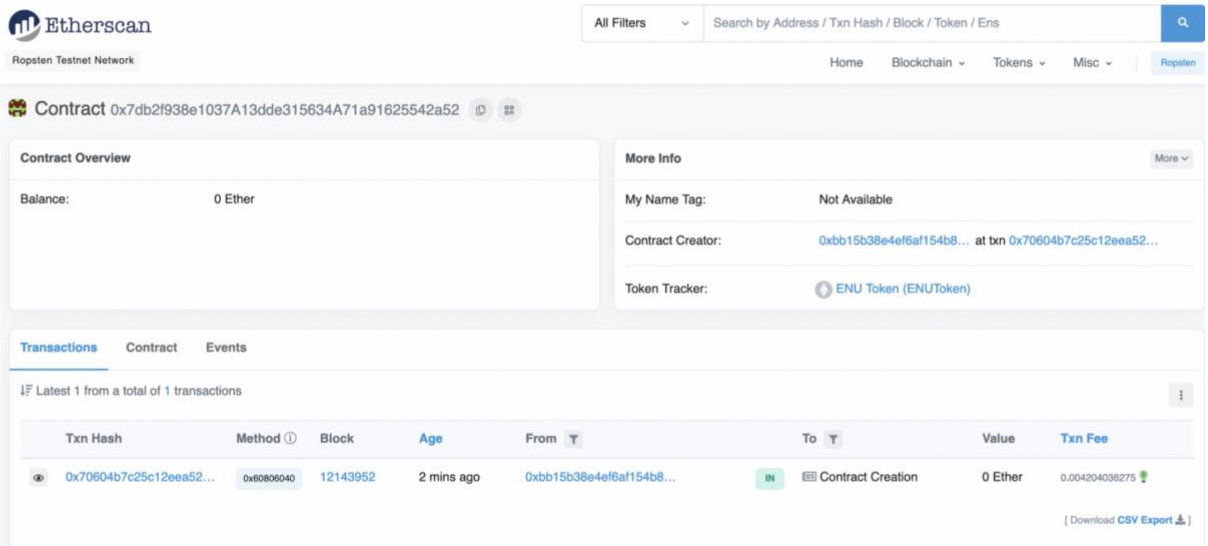
Status: Confirmed

From: 0xbB1...2233 → To: New Contract

Transaction

Nonce	45
Amount	-0 ETH
Gas Limit (Units)	4444686
Gas Used (Units)	1444686
Base Fee (GWEI)	0.000000082
Priority Fee (GWEI)	2.909999929
Total Gas Fee	0.004204 ETH
Max Fee Per Gas	0.000000003 ETH
Total	**0.00420404 ETH**

And then view the contact here]:



Contract 0x7db2f938e1037A13dde315634A71a91625542a52

Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

Contract Creator: 0xbb15b38e4ef6af154b8... at txn 0x70604b7c25c12eea52...

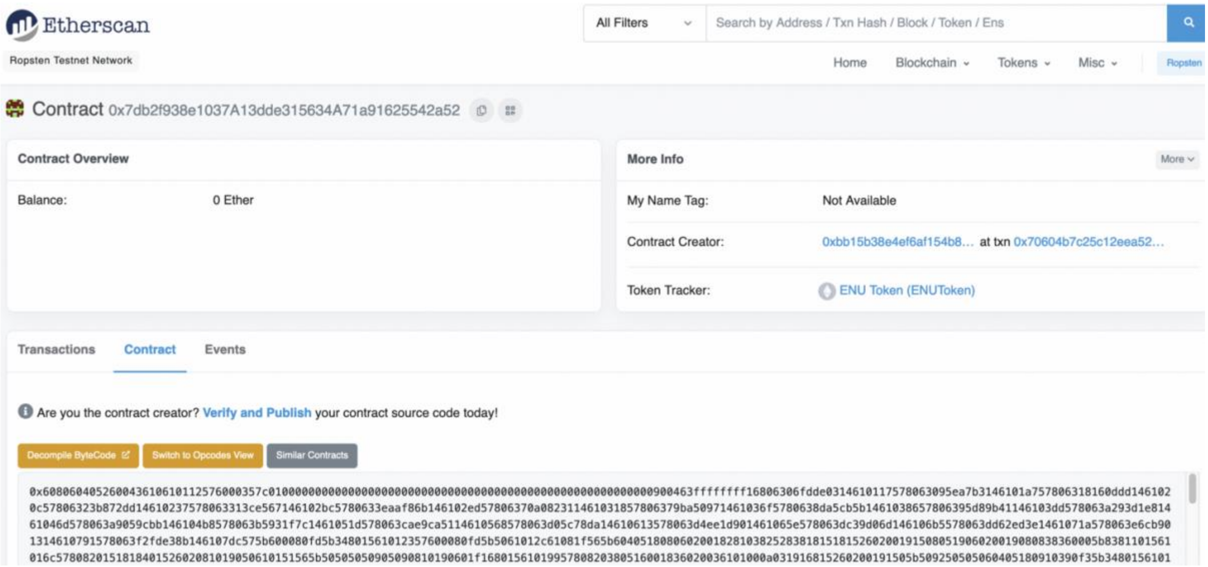
Token Tracker: ENU Token (ENUToken)

Latest 1 from a total of 1 transactions

	Txn Hash	Method	Block	Age	From		To	Value	Txn Fee
👁	0x70604b7c25c12eea52...	0x60806040	12143952	2 mins ago	0xbb15b38e4ef6af154b8...	IN	Contract Creation	0 Ether	0.004204036275

[Download CSV Export]

Next, we select the Contract tab:



Contract 0x7db2f938e1037A13dde315634A71a91625542a52

Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

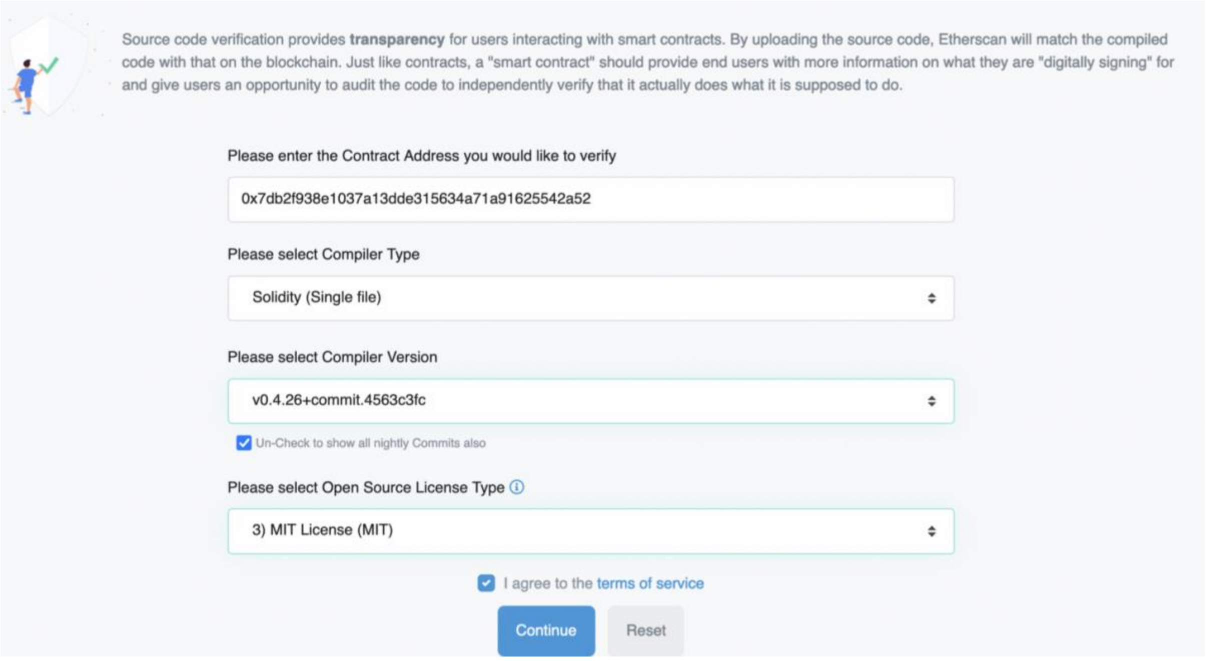
Contract Creator: 0xbb15b38e4ef6af154b8... at txn 0x70604b7c25c12eea52...

Token Tracker: ENU Token (ENUToken)

Are you the contract creator? Verify and Publish your contract source code today!

```
0x608060405260043610610112576000357c0100000000000000000000000000000000000000000000000000000000900463ffffffff16806306fdde031461011757806309 5ea7b3146101a757806318160ddd146102
0c57806323b872dd14610237578063313ce567146102bc5780633eaaf86b146102ed57806370a08231146103185 7806379ba50971461036f5780638da5cb5b1461038657806395d89b41146103dd578063a293d1e814
61046d578063a9059cbb146104b8578063b5931f7c1461051d578063cae9ca5114610568578063d05c78da14610613578063d4ee1d901461065e578063dc39d06d146106b5578063dd62ed3e1461071a578063e6cb90
1314610791578063f2fde38b146107dc575b600080fd5b348015610123576000 80fd5b5061012c61081f565b60405180806020018281038252838181518152602001915080519060200190808383600 05b8381101561
016c5780820151818401526020810190506101515 65b50505050905090810190601f1680156101995780820380516001836020036101000a031916815260200191505b509250505060405180910390f35b3480156101
```

And then select "Verify and Publish" and enter the details of the compiler version (v0.4.26):



Source code verification provides **transparency** for users interacting with smart contracts. By uploading the source code, Etherscan will match the compiled code with that on the blockchain. Just like contracts, a "smart contract" should provide end users with more information on what they are "digitally signing" for and give users an opportunity to audit the code to independently verify that it actually does what it is supposed to do.

Please enter the Contract Address you would like to verify

0x7db2f938e1037a13dde315634a71a91625542a52

Please select Compiler Type

Solidity (Single file)

Please select Compiler Version

v0.4.26+commit.4563c3fc

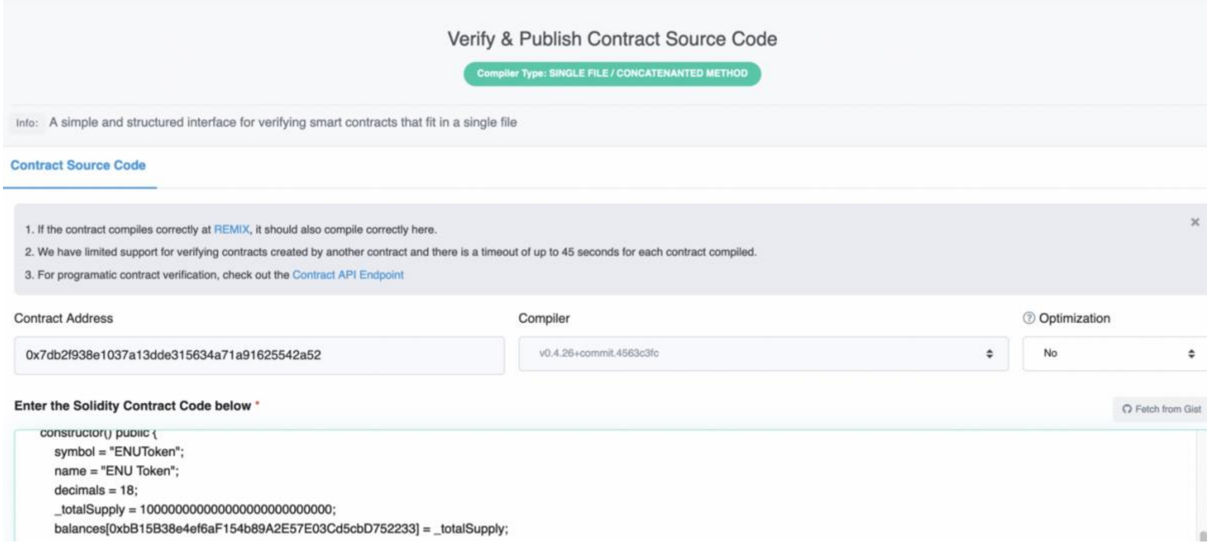
☑ Un-Check to show all nightly Commits also

Please select Open Source License Type

3) MIT License (MIT)

☑ I agree to the terms of service

We then need to copy-and-pasete the contract code into the Source Code text box:



Verify & Publish Contract Source Code

Compiler Type: SINGLE FILE / CONCATENANTED METHOD

Info: A simple and structured interface for verifying smart contracts that fit in a single file

Contract Source Code

1. If the contract compiles correctly at REMIX, it should also compile correctly here.
2. We have limited support for verifying contracts created by another contract and there is a timeout of up to 45 seconds for each contract compiled.
3. For programatic contract verification, check out the Contract API Endpoint

Contract Address: 0x7db2f938e1037a13dde315634a71a91625542a52

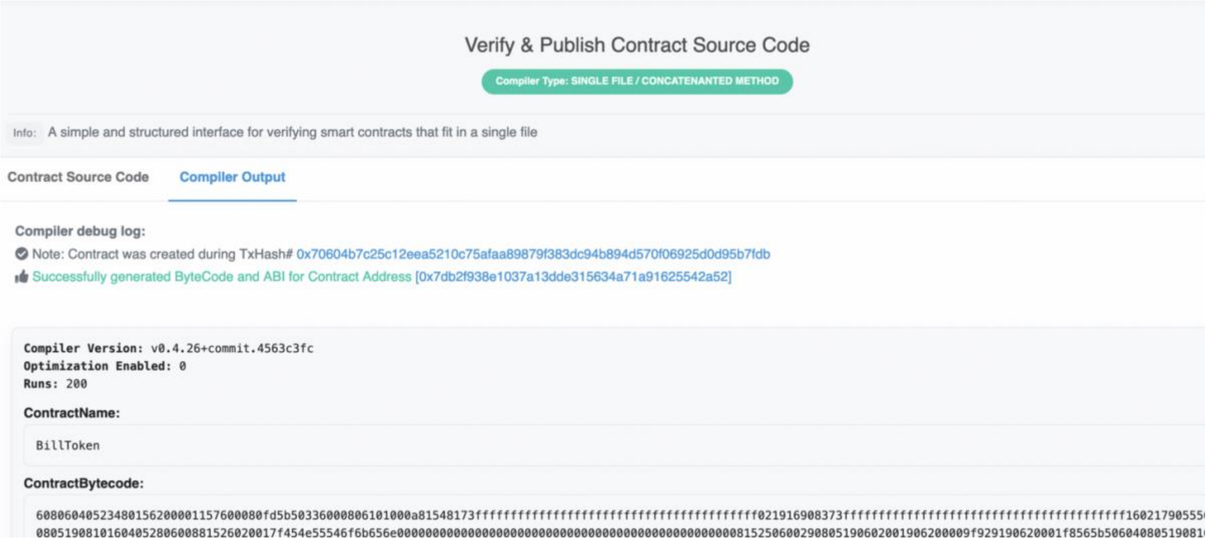
Compiler: v0.4.26+commit.4563c3fc

Optimization: No

Enter the Solidity Contract Code below *

```
    constructor() public {
        symbol = "ENUToken";
        name = "ENU Token";
        decimals = 18;
        _totalSupply = 1000000000000000000000000000;
        balances[0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233] = _totalSupply;
```

After less than 45 seconds, the contract will be approved:



Verify & Publish Contract Source Code

Compiler Type: SINGLE FILE / CONCATENANTED METHOD

Info: A simple and structured interface for verifying smart contracts that fit in a single file

Compiler debug log:

Note: Contract was created during TxHash# 0x70604b7c25c12eea5210c75afaa89879f383dc94b894d570f06925d0d95b7fdb

Successfully generated ByteCode and ABI for Contract Address [0x7db2f938e1037a13dde315634a71a91625542a52]

```
Compiler Version: v0.4.26+commit.4563c3fc
Optimization Enabled: 0
Runs: 200
```

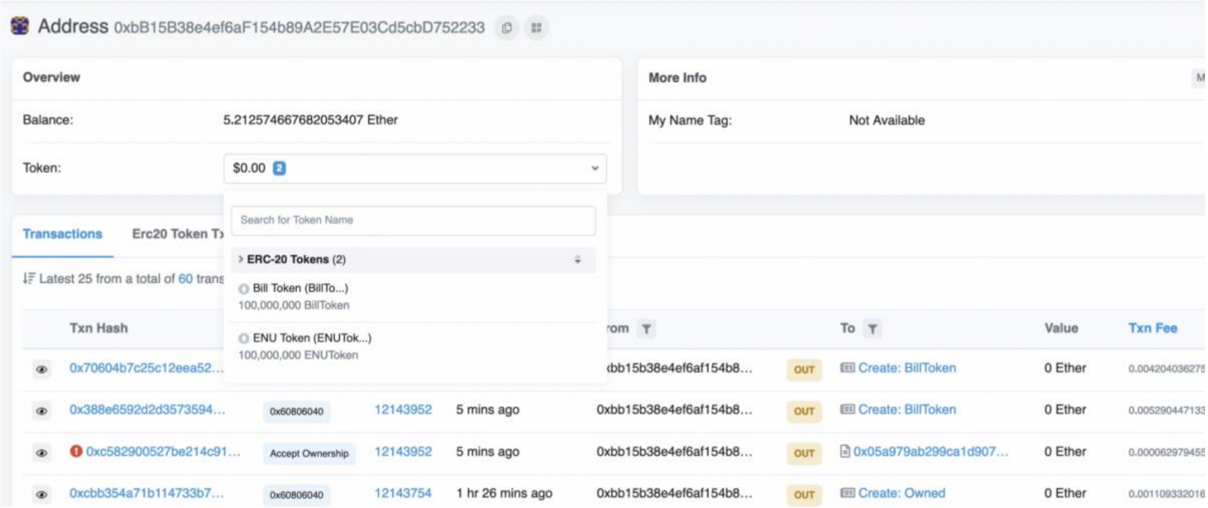
ContractName:

```
BillToken
```

ContractBytecode:

```
608060405234801562000011576000080fd5b50336000806101000a81548173ffffffffffffffffffffffffffffffffffffffff021916908373ffffffffffffffffffffffffffffffffffffffff16021790555
0805190810160405280600881526020017f454e55546f6b656e000000000000000000000000000000000000000000000000815250600290805190602001906200009f929190620001f8565b50604080519081
```

When the contact is run there is a constructor to transfer the tokens to the wallet we have defined (and who will be the owner of the token). We can now go back to the wallet which is specified, to see if the tokens have been transferred:



Address 0xbB15B38e4ef6aF154b89A2E57E03Cd5cbD752233

Overview	
Balance:	5.212574667682053407 Ether
Token:	$0.00 2

More Info	
My Name Tag:	Not Available

Search for Token Name

ERC-20 Tokens (2)

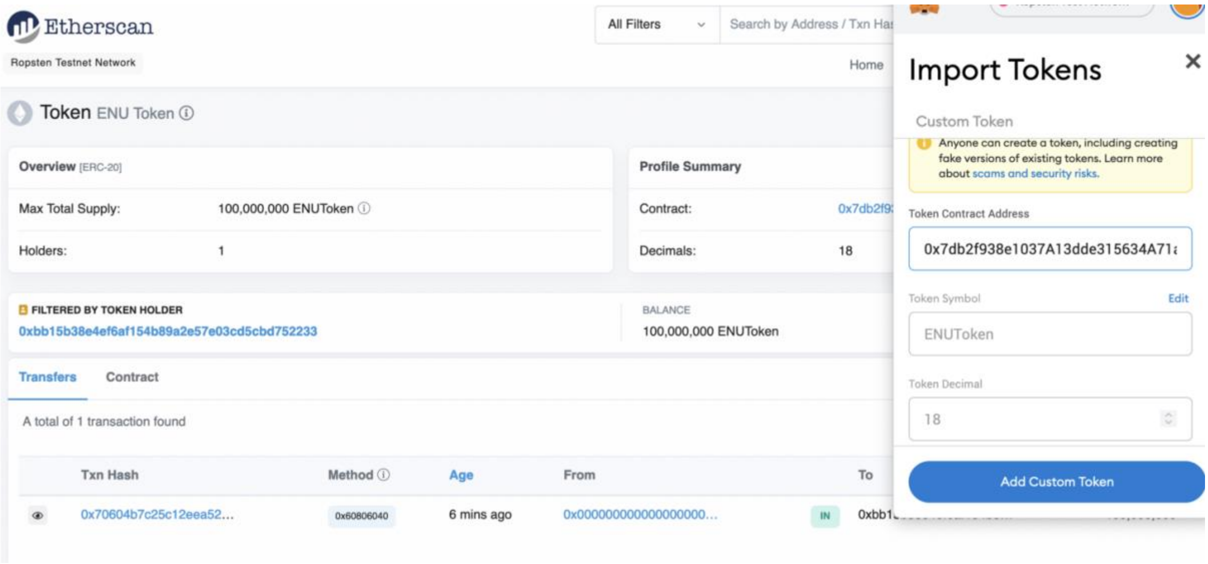
Bill Token (BillTo...)
100,000,000 BillToken

ENU Token (ENUTok...)
100,000,000 ENUToken

Latest 25 from a total of 60 trans

Txn Hash				rom		To	Value	Txn Fee
0x70604b7c25c12eea52…				xbb15b38e4ef6af154b8…	OUT	Create: BillToken	0 Ether	0.004204036275
0x388e6592d2d3573594…	0x60806040	12143952	5 mins ago	0xbb15b38e4ef6af154b8…	OUT	Create: BillToken	0 Ether	0.005290447133
0xc582900527be214c91…	Accept Ownership	12143952	5 mins ago	0xbb15b38e4ef6af154b8…	OUT	0x05a979ab299ca1d907…	0 Ether	0.000062979455
0xcbb354a71b114733b7…	0x60806040	12143754	1 hr 26 mins ago	0xbb15b38e4ef6af154b8…	OUT	Create: Owned	0 Ether	0.001109332016

Next, we can transfer the tokens into our wallet, by defining the contract address:


Etherscan

Ropsten Testnet Network

All Filters

Search by Address / Txn Has

Home

Token ENU Token

Overview [ERC-20]

Max Total Supply: 100,000,000 ENUToken

Holders: 1

Profile Summary

Contract: 0x7db2f9

Decimals: 18

FILTERED BY TOKEN HOLDER
0xbb15b38e4ef6af154b89a2e57e03cd5cbd752233

BALANCE
100,000,000 ENUToken

Transfers | Contract

A total of 1 transaction found

	Txn Hash	Method	Age	From		To
👁	0x70604b7c25c12eea52...	0x60806040	6 mins ago	0x000000000000000000...	IN	0xbb1

Import Tokens ✕

Custom Token

Anyone can create a token, including creating fake versions of existing tokens. Learn more about scams and security risks.

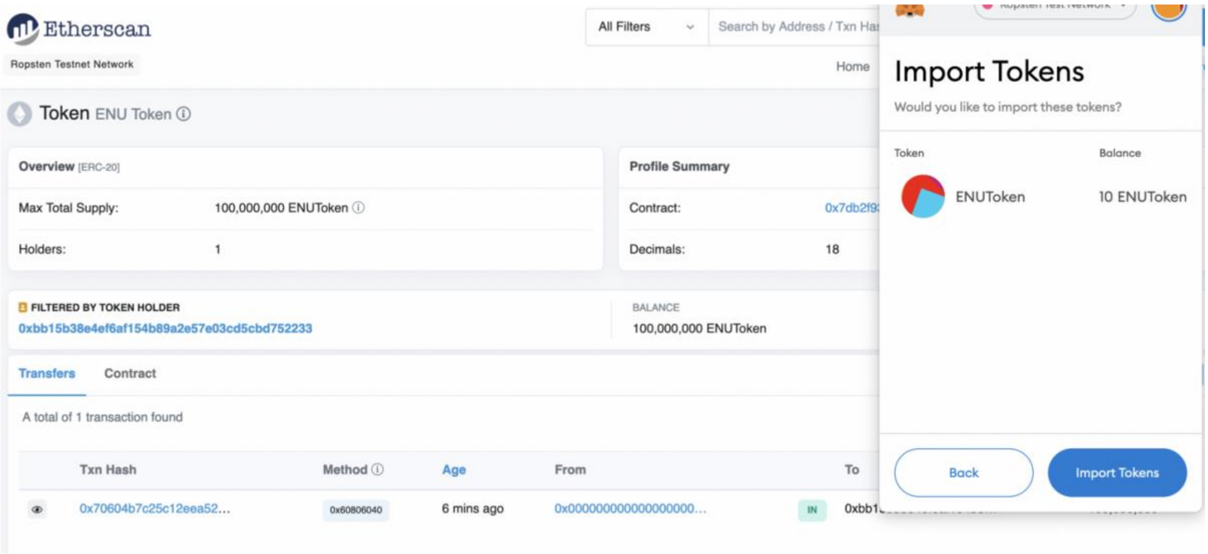
Token Contract Address: 0x7db2f938e1037A13dde315634A71a

Token Symbol (Edit): ENUToken

Token Decimal: 18

Add Custom Token

We will now have our new tokens in the wallet:


Etherscan

Ropsten Testnet Network

All Filters

Search by Address / Txn Has

Home

Token ENU Token

Overview [ERC-20]

Max Total Supply: 100,000,000 ENUToken

Holders: 1

Profile Summary

Contract: 0x7db2f9

Decimals: 18

FILTERED BY TOKEN HOLDER
0xbb15b38e4ef6af154b89a2e57e03cd5cbd752233

BALANCE
100,000,000 ENUToken

Transfers | Contract

A total of 1 transaction found

	Txn Hash	Method	Age	From		To
👁	0x70604b7c25c12eea52...	0x60806040	6 mins ago	0x000000000000000000...	IN	0xbb1

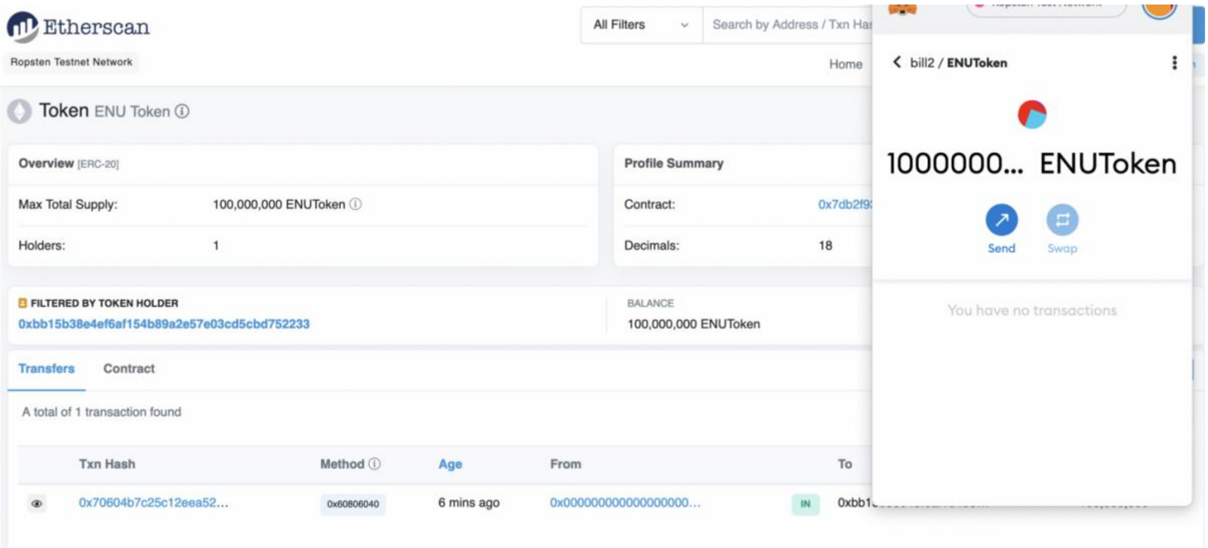
Import Tokens

Would you like to import these tokens?

Token	Balance
ENUToken	10 ENUToken

Back | Import Tokens

And with:


Etherscan

Ropsten Testnet Network

All Filters | Search by Address / Txn Has

Home

Token ENU Token

Overview [ERC-20]

Max Total Supply: 100,000,000 ENUToken

Holders: 1

Profile Summary

Contract: 0x7db2f9

Decimals: 18

FILTERED BY TOKEN HOLDER

0xbb15b38e4ef6af154b89a2e57e03cd5cbd752233

BALANCE

100,000,000 ENUToken

Transfers | Contract

A total of 1 transaction found

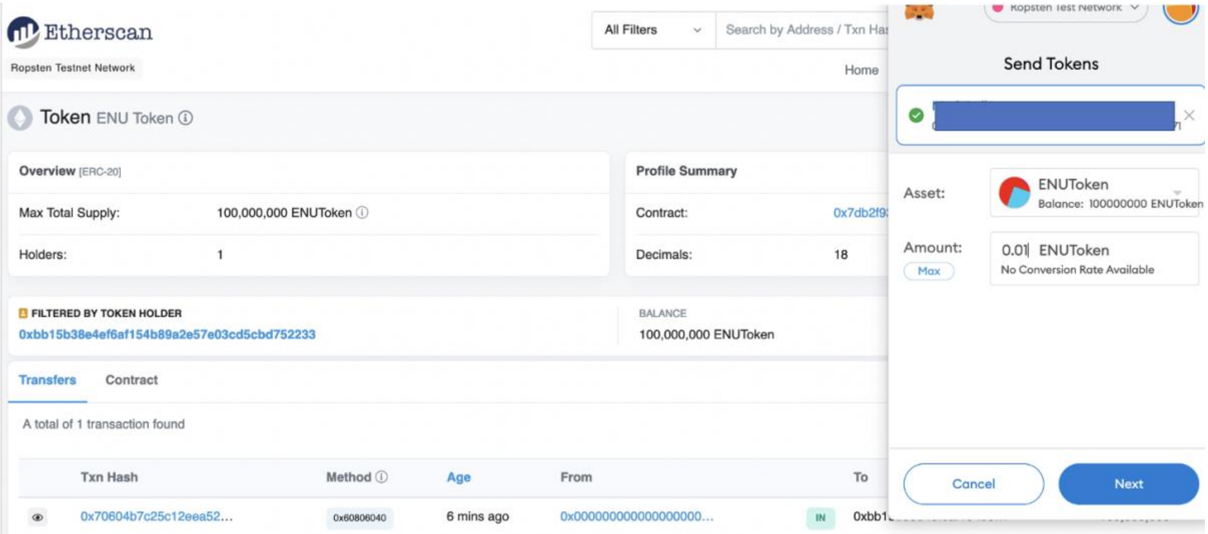
	Txn Hash	Method	Age	From		To
👁	0x70604b7c25c12eea52...	0x60806040	6 mins ago	0x000000000000000000...	IN	0xbb1

bill2 / ENUToken

1000000... ENUToken

You have no transactions

We can now transfer the cryptocurrency to another wallet:


Etherscan

Ropsten Testnet Network

All Filters | Search by Address / Txn Has

Home

Token ENU Token

Overview [ERC-20]

Max Total Supply: 100,000,000 ENUToken

Holders: 1

Profile Summary

Contract: 0x7db2f9

Decimals: 18

FILTERED BY TOKEN HOLDER

0xbb15b38e4ef6af154b89a2e57e03cd5cbd752233

BALANCE

100,000,000 ENUToken

Transfers | Contract

A total of 1 transaction found

	Txn Hash	Method	Age	From		To
👁	0x70604b7c25c12eea52...	0x60806040	6 mins ago	0x000000000000000000...	IN	0xbb1

Ropsten Test Network

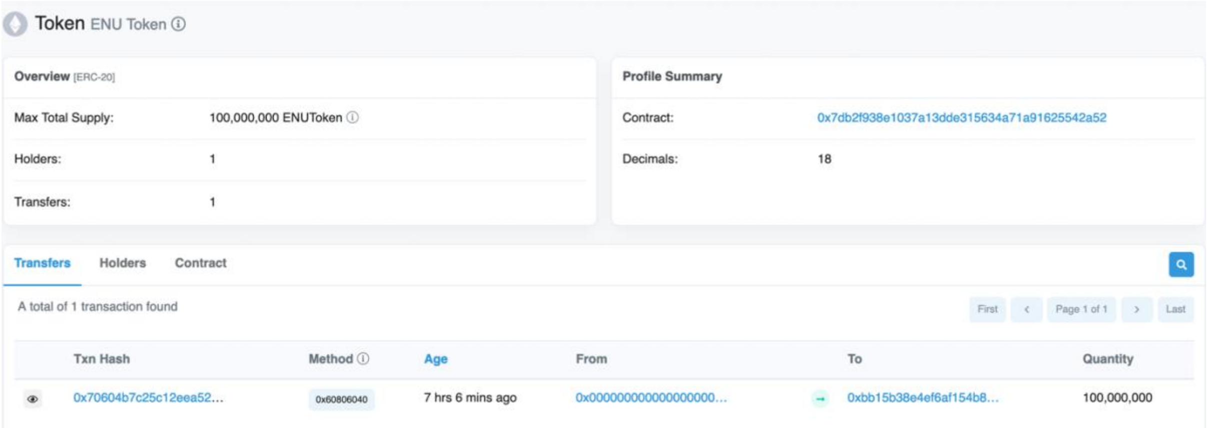
Send Tokens

✓ [redacted]

Asset: ENUToken Balance: 100000000 ENUToken

Amount: 0.01 ENUToken No Conversion Rate Available

We can view the ENUToken: here]:


Token ENU Token

Overview [ERC-20]

Max Total Supply: 100,000,000 ENUToken

Holders: 1

Transfers: 1

Profile Summary

Contract: 0x7db2f938e1037a13dde315634a71a91625542a52

Decimals: 18

Transfers | Holders | Contract

A total of 1 transaction found

	Txn Hash	Method	Age	From		To	Quantity
👁	0x70604b7c25c12eea52...	0x60806040	7 hrs 6 mins ago	0x000000000000000000...	→	0xbb15b38e4ef6af154b8...	100,000,000

Now answer the following:

- Do you see the tokens in your wallet?
- Now send 0.1 of your token to someone else's wallet. If you want, you can send to your tutor's wallet. Bill's wallet is 0xbb15b38e4ef6af154b89a2e57e03cd5cbd752233
- Did they receive the token?