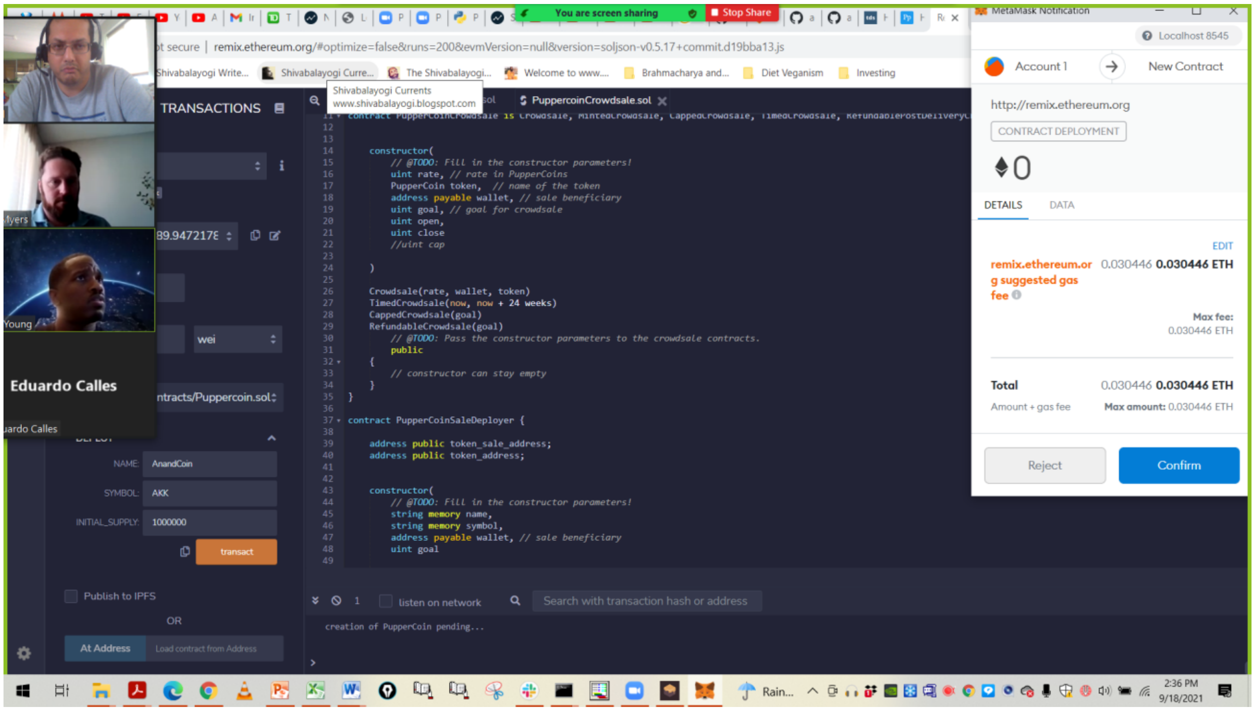
Deploying the pupper coin crowd sale contract, you put in the name and symbol of your coin. In my case I called it AnandCoin with a symbol of AKK and a supply of 1000000. I hit transact and confirm on the metamask and this is the deployed contract:



remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js

PuppercoinCrowdsale.sol

```
contract PupperCoinCrowdsale is Crowdsale, MintedCrowdsale, CappedCrowdsale, TimedCrowdsale, RefundablePostDeliveryC


    constructor(
        // @TODO: Fill in the constructor parameters!
        uint rate, // rate in PupperCoins
        PupperCoin token,  // name of the token
        address payable wallet, // sale beneficiary
        uint goal, // goal for crowdsale
        uint open,
        uint close
        //uint cap

    )

    Crowdsale(rate, wallet, token)
    TimedCrowdsale(now, now + 24 weeks)
    CappedCrowdsale(goal)
    RefundableCrowdsale(goal)
        // @TODO: Pass the constructor parameters to the crowdsale contracts.
        public
    {
        // constructor can stay empty
    }
}

contract PupperCoinSaleDeployer {

    address public token_sale_address;
    address public token_address;


    constructor(
        // @TODO: Fill in the constructor parameters!
        string memory name,
        string memory symbol,
        address payable wallet, // sale beneficiary
        uint goal

```

DEPLOY

NAME: AnandCoin

SYMBOL: AKK

INITIAL_SUPPLY: 1000000

transact

Publish to IPFS

OR

At Address | Load contract from Address

```
creation of PupperCoin pending...
```

MetaMask Notification

Localhost 8545

Account 1 → New Contract

http://remix.ethereum.org

CONTRACT DEPLOYMENT

0

DETAILS | DATA

EDIT

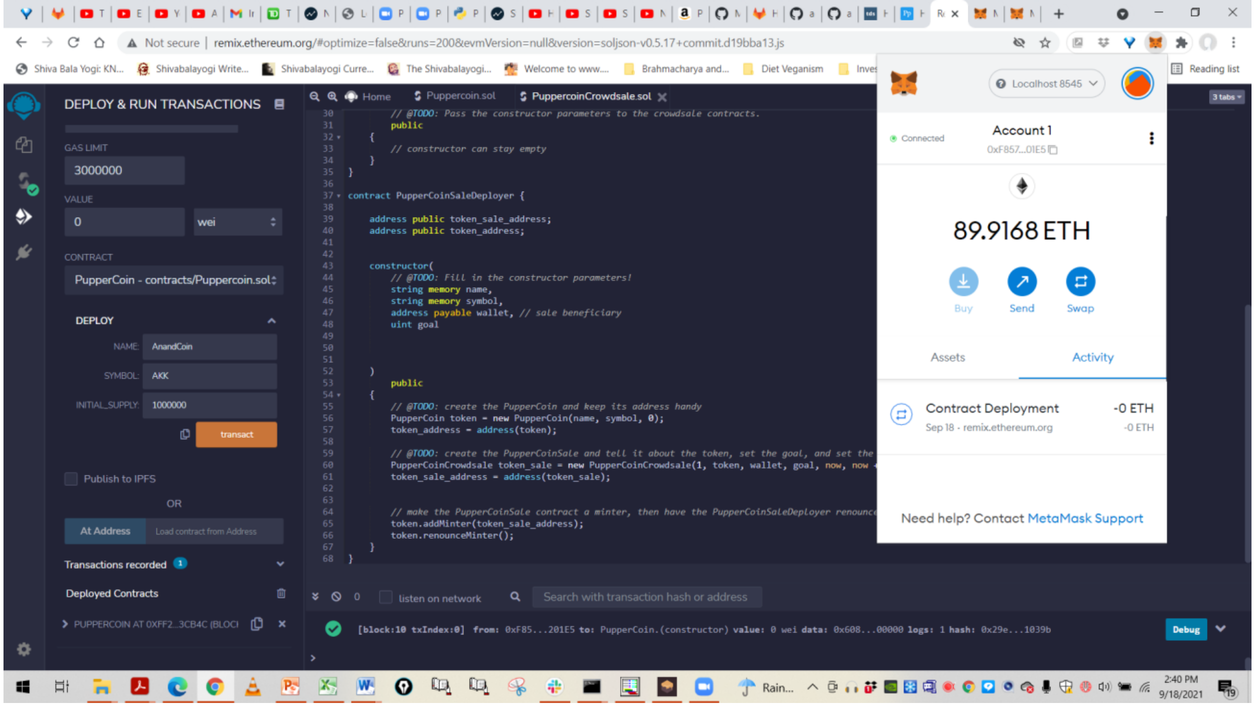
remix.ethereum.org suggested gas fee: 0.030446 **0.030446 ETH**

Max fee: 0.030446 ETH

Total: 0.030446 **0.030446 ETH**

Amount + gas fee — **Max amount:** 0.030446 ETH

Reject | Confirm



remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js

DEPLOY & RUN TRANSACTIONS

GAS LIMIT: 3000000

VALUE: 0 wei

CONTRACT: PupperCoin - contracts/Puppercoin.sol

DEPLOY

NAME: AnandCoin

SYMBOL: AKK

INITIAL_SUPPLY: 1000000

transact

Publish to IPFS

OR

At Address | Load contract from Address

Transactions recorded 1

Deployed Contracts

PUPPERCOIN AT 0XFF2...3CB4C (BLOC

Home | Puppercoin.sol | PuppercoinCrowdsale.sol

```
        // @TODO: Pass the constructor parameters to the crowdsale contracts.
        public
    {
        // constructor can stay empty
    }
}

contract PupperCoinSaleDeployer {

    address public token_sale_address;
    address public token_address;


    constructor(
        // @TODO: Fill in the constructor parameters!
        string memory name,
        string memory symbol,
        address payable wallet, // sale beneficiary
        uint goal



    )
        public
    {
        // @TODO: create the PupperCoin and keep its address handy
        PupperCoin token = new PupperCoin(name, symbol, 0);
        token_address = address(token);

        // @TODO: create the PupperCoinSale and tell it about the token, set the goal, and set the
        PupperCoinCrowdsale token_sale = new PupperCoinCrowdsale(1, token, wallet, goal, now, now
        token_sale_address = address(token_sale);


        // make the PupperCoinSale contract a minter, then have the PupperCoinSaleDeployer renounce
        token.addMinter(token_sale_address);
        token.renounceMinter();
    }
}
```

```
[block:10 txIndex:0] from: 0xF85...201E5 to: PupperCoin.(constructor) value: 0 wei data: 0x608...00000 logs: 1 hash: 0x29e...1039b
```

MetaMask

Localhost 8545

Connected — Account 1 — 0xF857...01E5

89.9168 ETH

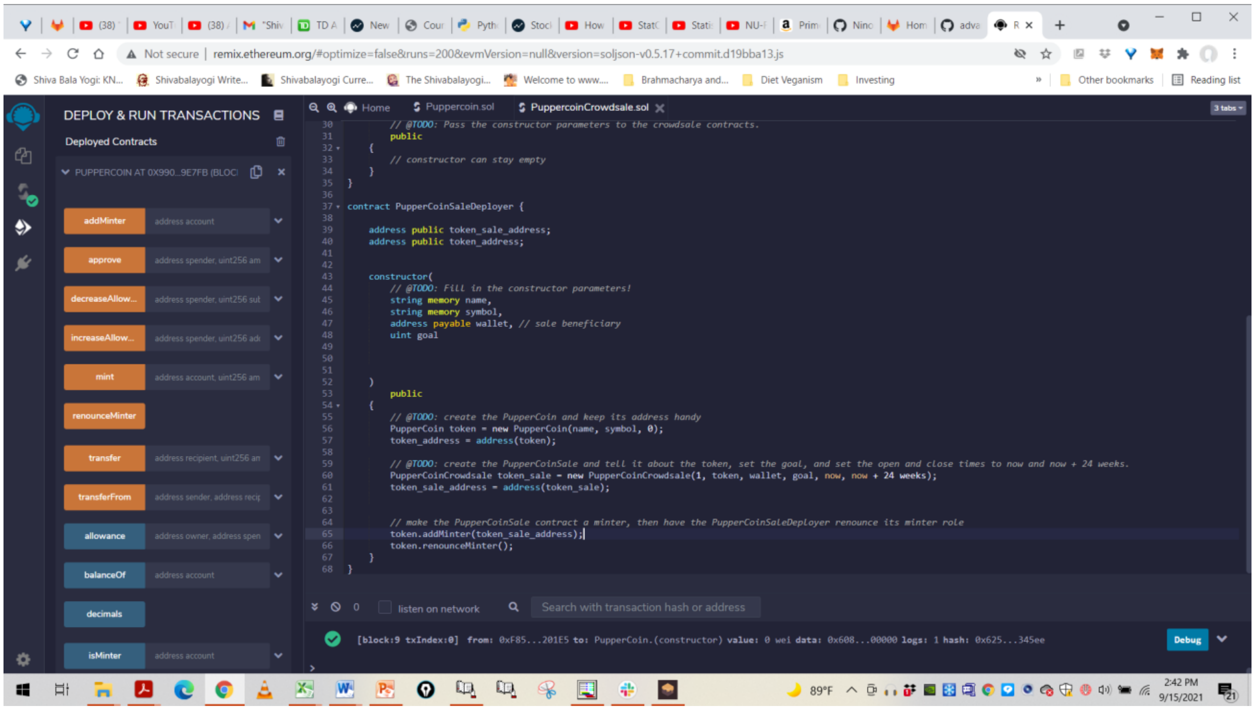
Buy | Send | Swap

Assets | Activity

Contract Deployment — -0 ETH

Sep 18 - remix.ethereum.org — -0 ETH

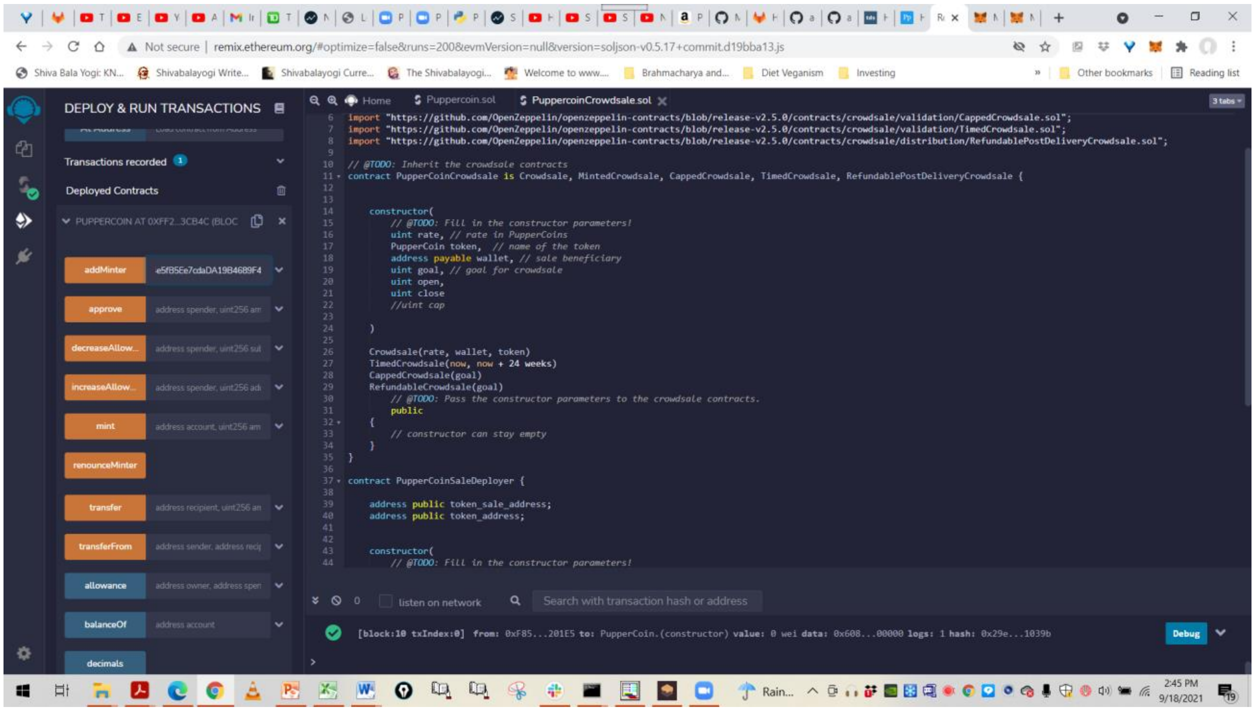
Need help? Contact MetaMask Support



```
        // @TODO: Pass the constructor parameters to the crowdsale contracts.
        public
    {
        // constructor can stay empty
    }
}

contract PupperCoinSaleDeployer {

    address public token_sale_address;
    address public token_address;


    constructor(
        // @TODO: Fill in the constructor parameters!
        string memory name,
        string memory symbol,
        address payable wallet, // sale beneficiary
        uint goal



    )
        public
    {
        // @TODO: create the PupperCoin and keep its address handy
        PupperCoin token = new PupperCoin(name, symbol, 0);
        token_address = address(token);

        // @TODO: create the PupperCoinSale and tell it about the token, set the goal, and set the open and close times to now and now + 24 weeks.
        PupperCoinCrowdsale token_sale = new PupperCoinCrowdsale(1, token, wallet, goal, now, now + 24 weeks);
        token_sale_address = address(token_sale);


        // make the PupperCoinSale contract a minter, then have the PupperCoinSaleDeployer renounce its minter role
        token.addMinter(token_sale_address);
        token.renounceMinter();
    }
}
```

```
[block:9 txIndex:0] from: 0xF85...201E5 to: PupperCoin.(constructor) value: 0 wei data: 0x608...00000 logs: 1 hash: 0x625...345ee
```

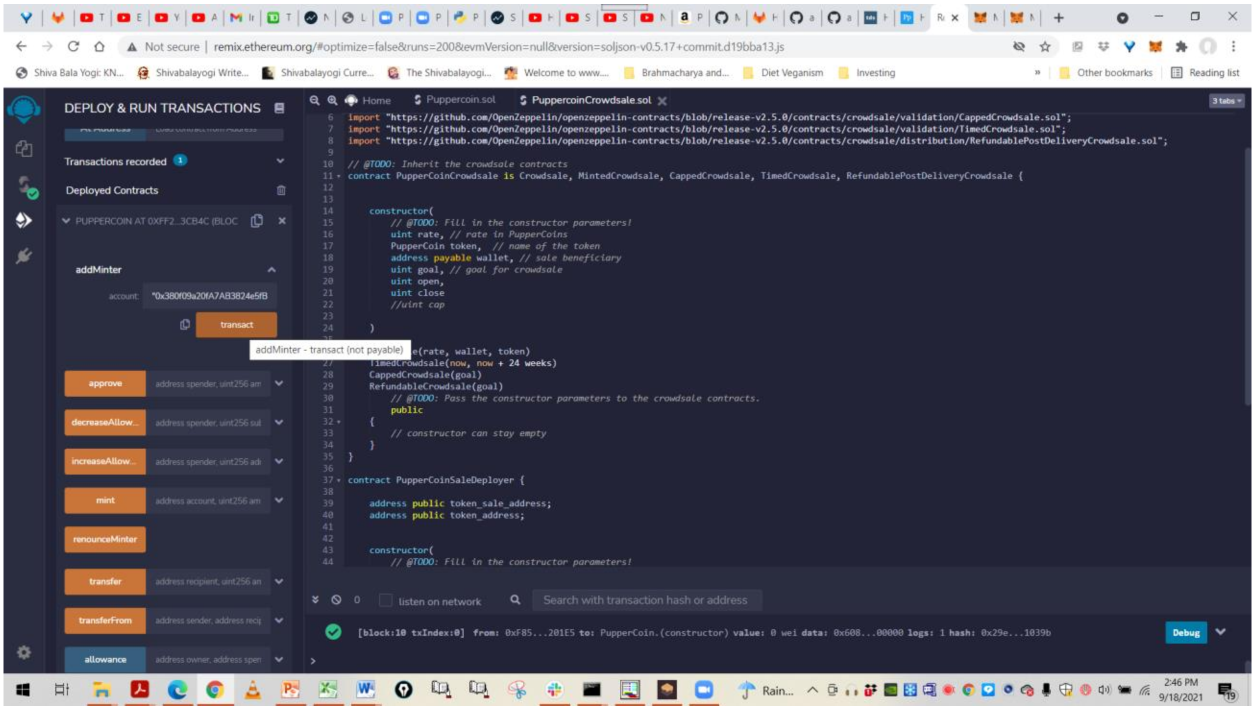
Add minter:



addMinter: e5f85Ee7cdaDA19B4689F4

```
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/validation/CappedCrowdsale.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/validation/TimedCrowdsale.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/distribution/RefundablePostDeliveryCrowdsale.sol";

// @TODO: Inherit the crowdsale contracts
contract PupperCoinCrowdsale is Crowdsale, MintedCrowdsale, CappedCrowdsale, TimedCrowdsale, RefundablePostDeliveryCrowdsale {


    constructor(
        // @TODO: Fill in the constructor parameters!
        uint rate, // rate in PupperCoins
        PupperCoin token,  // name of the token
        address payable wallet, // sale beneficiary
        uint goal, // goal for crowdsale
        uint open,
        uint close
        //uint cap

    )

    Crowdsale(rate, wallet, token)
    TimedCrowdsale(now, now + 24 weeks)
    CappedCrowdsale(goal)
    RefundableCrowdsale(goal)
        // @TODO: Pass the constructor parameters to the crowdsale contracts.
        public
    {
        // constructor can stay empty
    }
}

contract PupperCoinSaleDeployer {

    address public token_sale_address;
    address public token_address;


    constructor(
        // @TODO: Fill in the constructor parameters!
```

```
[block:10 txIndex:0] from: 0xF85...201E5 to: PupperCoin.(constructor) value: 0 wei data: 0x608...00000 logs: 1 hash: 0x29e...1039b
```



Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js

DEPLOY & RUN TRANSACTIONS

Transactions recorded 1

Deployed Contracts

PUPPERCOIN AT 0XFF2...3CB4C (BLOC

addMinter

account: "0x380f09a20fA7AB3824e5fB

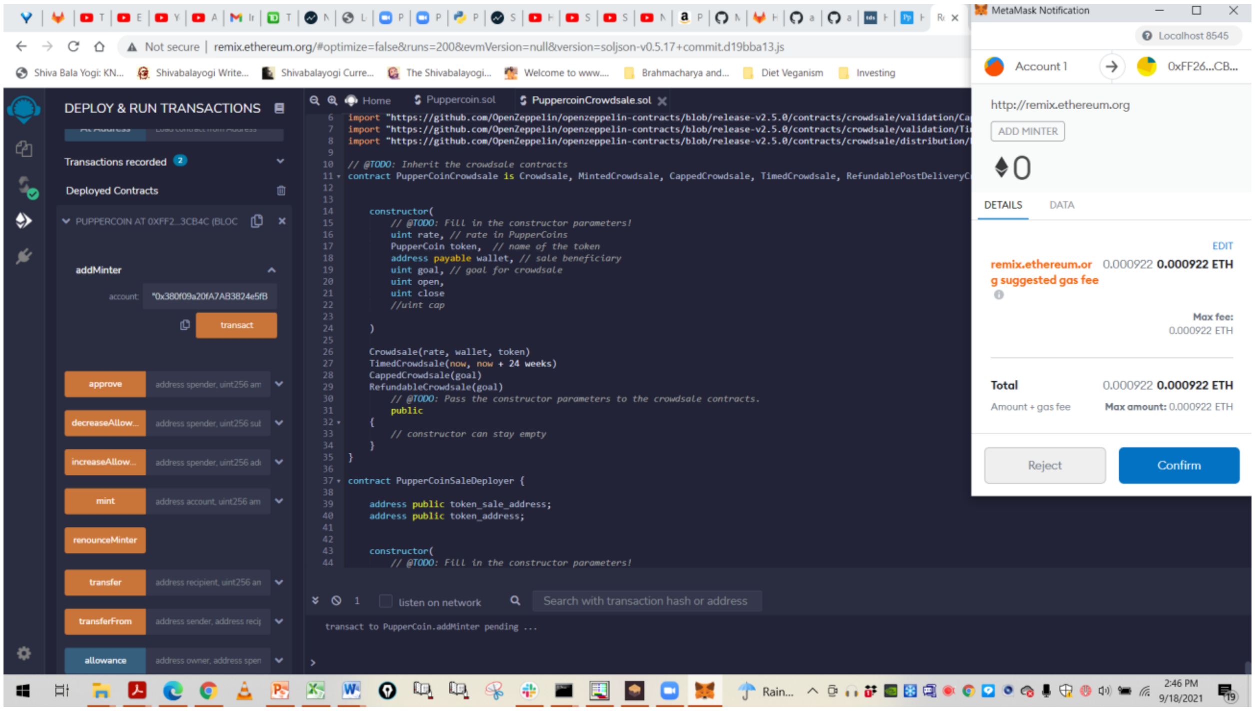
transact

addMinter - transact (not payable)

approve
address spender, uint256 am
decreaseAllow...
address spender, uint256 sul
increaseAllow...
address spender, uint256 ad
mint
address account, uint256 am
renounceMinter
transfer
address recipient, uint256 an
transferFrom
address sender, address reci
allowance
address owner, address spen
Home
Puppercoin.sol
PuppercoinCrowdsale.sol
3 tabs
listen on network
Search with transaction hash or address
Debug
Rain...
2:46 PM
9/18/2021

```
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/validation/CappedCrowdsale.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/validation/TimedCrowdsale.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/distribution/RefundablePostDeliveryCrowdsale.sol";

// @TODO: Inherit the crowdsale contracts
contract PupperCoinCrowdsale is Crowdsale, MintedCrowdsale, CappedCrowdsale, TimedCrowdsale, RefundablePostDeliveryCrowdsale {


    constructor(
        // @TODO: Fill in the constructor parameters!
        uint rate, // rate in PupperCoins
        PupperCoin token,  // name of the token
        address payable wallet, // sale beneficiary
        uint goal, // goal for crowdsale
        uint open,
        uint close
        //uint cap

    )

              e(rate, wallet, token)
    TimedCrowdsale(now, now + 24 weeks)
    CappedCrowdsale(goal)
    RefundableCrowdsale(goal)
        // @TODO: Pass the constructor parameters to the crowdsale contracts.
        public
    {
        // constructor can stay empty
    }
}

contract PupperCoinSaleDeployer {

    address public token_sale_address;
    address public token_address;


    constructor(
        // @TODO: Fill in the constructor parameters!
```

```
0
[block:10 txIndex:0] from: 0xF85...201E5 to: PupperCoin.(constructor) value: 0 wei data: 0x608...00000 logs: 1 hash: 0x29e...1039b
>
```



Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js

DEPLOY & RUN TRANSACTIONS

Transactions recorded 2

Deployed Contracts

PUPPERCOIN AT 0XFF2...3CB4C (BLOC

addMinter

account: "0x380f09a20fA7AB3824e5fB

transact

approve
address spender, uint256 am
decreaseAllow...
address spender, uint256 sul
increaseAllow...
address spender, uint256 ad
mint
address account, uint256 am
renounceMinter
transfer
address recipient, uint256 an
transferFrom
address sender, address reci
allowance
address owner, address spen
Home
Puppercoin.sol
PuppercoinCrowdsale.sol
listen on network
Search with transaction hash or address
MetaMask Notification
Localhost 8545
DETAILS
DATA
EDIT
Reject
Confirm
Rain...
2:46 PM
9/18/2021

```
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/validation/Ca
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/validation/Ti
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/distribution/

// @TODO: Inherit the crowdsale contracts
contract PupperCoinCrowdsale is Crowdsale, MintedCrowdsale, CappedCrowdsale, TimedCrowdsale, RefundablePostDeliveryC


    constructor(
        // @TODO: Fill in the constructor parameters!
        uint rate, // rate in PupperCoins
        PupperCoin token,  // name of the token
        address payable wallet, // sale beneficiary
        uint goal, // goal for crowdsale
        uint open,
        uint close
        //uint cap

    )

    Crowdsale(rate, wallet, token)
    TimedCrowdsale(now, now + 24 weeks)
    CappedCrowdsale(goal)
    RefundableCrowdsale(goal)
        // @TODO: Pass the constructor parameters to the crowdsale contracts.
        public
    {
        // constructor can stay empty
    }
}

contract PupperCoinSaleDeployer {

    address public token_sale_address;
    address public token_address;


    constructor(
        // @TODO: Fill in the constructor parameters!
```

```
1
transact to PupperCoin.addMinter pending ...
>
```

Account 1 → 0xFF26...CB...

http://remix.ethereum.org

ADD MINTER

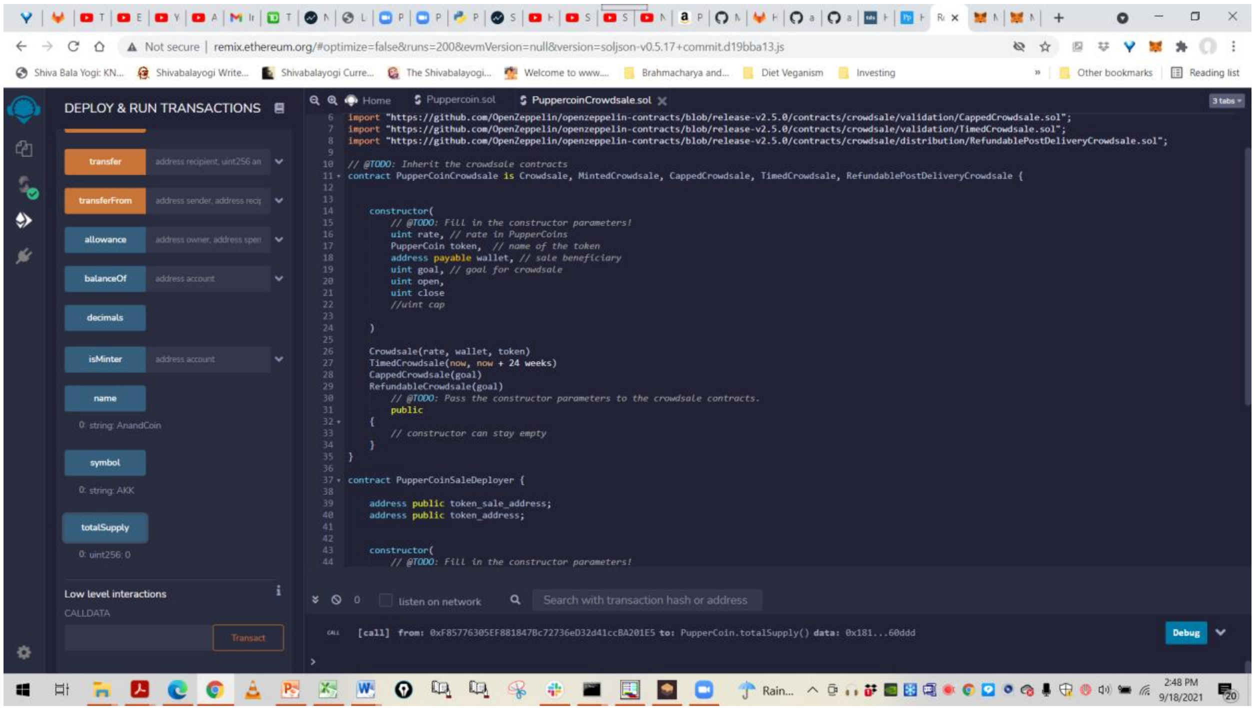
♦0

remix.ethereum.org suggested gas fee 0.000922 **0.000922 ETH**

Max fee: 0.000922 ETH

Total 0.000922 **0.000922 ETH**

Amount + gas fee **Max amount:** 0.000922 ETH



Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js

name — 0: string: AnandCoin

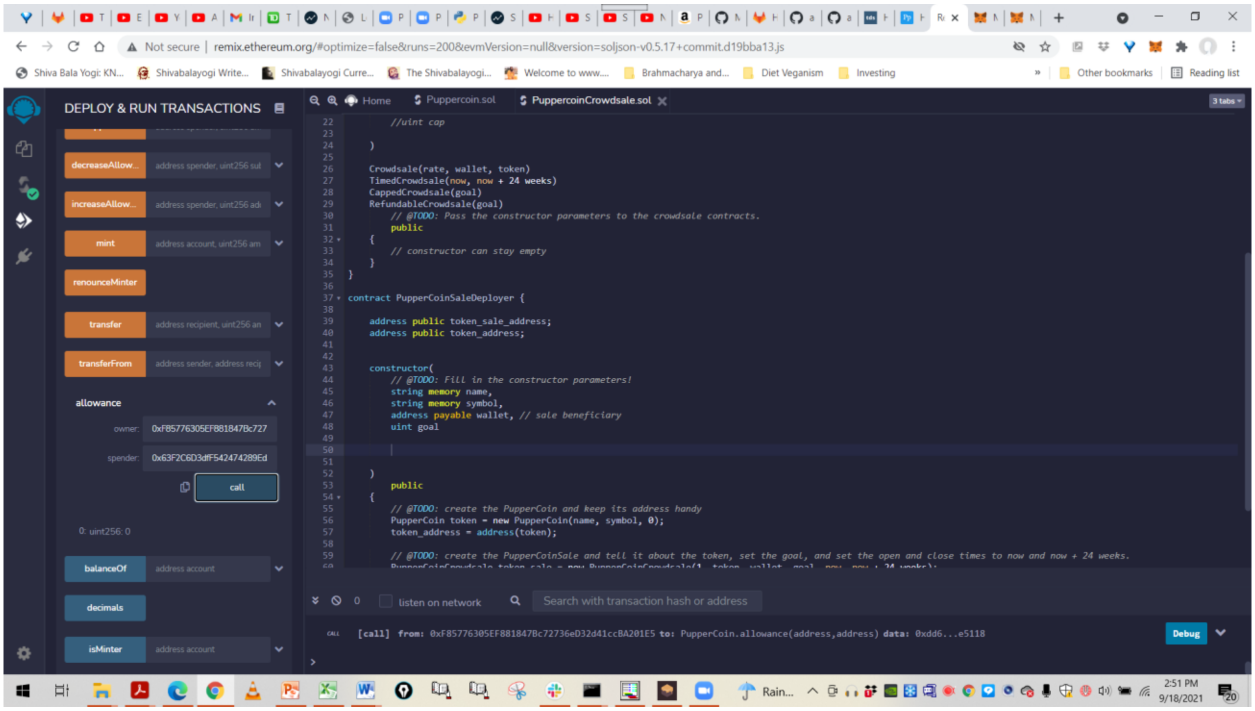
symbol — 0: string: AKK

totalSupply — 0: uint256: 0

```
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/validation/CappedCrowdsale.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/validation/TimedCrowdsale.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/crowdsale/distribution/RefundablePostDeliveryCrowdsale.sol";

// @TODO: Inherit the crowdsale contracts
contract PupperCoinCrowdsale is Crowdsale, MintedCrowdsale, CappedCrowdsale, TimedCrowdsale, RefundablePostDeliveryCrowdsale {


    constructor(
        // @TODO: Fill in the constructor parameters!
        uint rate, // rate in PupperCoins
        PupperCoin token,  // name of the token
        address payable wallet, // sale beneficiary
        uint goal, // goal for crowdsale
        uint open,
        uint close
        //uint cap

    )

    Crowdsale(rate, wallet, token)
    TimedCrowdsale(now, now + 24 weeks)
    CappedCrowdsale(goal)
    RefundableCrowdsale(goal)
        // @TODO: Pass the constructor parameters to the crowdsale contracts.
        public
    {
        // constructor can stay empty
    }
}

contract PupperCoinSaleDeployer {

    address public token_sale_address;
    address public token_address;


    constructor(
        // @TODO: Fill in the constructor parameters!
```

```
[call] from: 0xF85776305EF881847Bc72736eD32d41ccBA201E5 to: PupperCoin.totalSupply() data: 0x181...60ddd
```

2:48 PM
9/18/2021



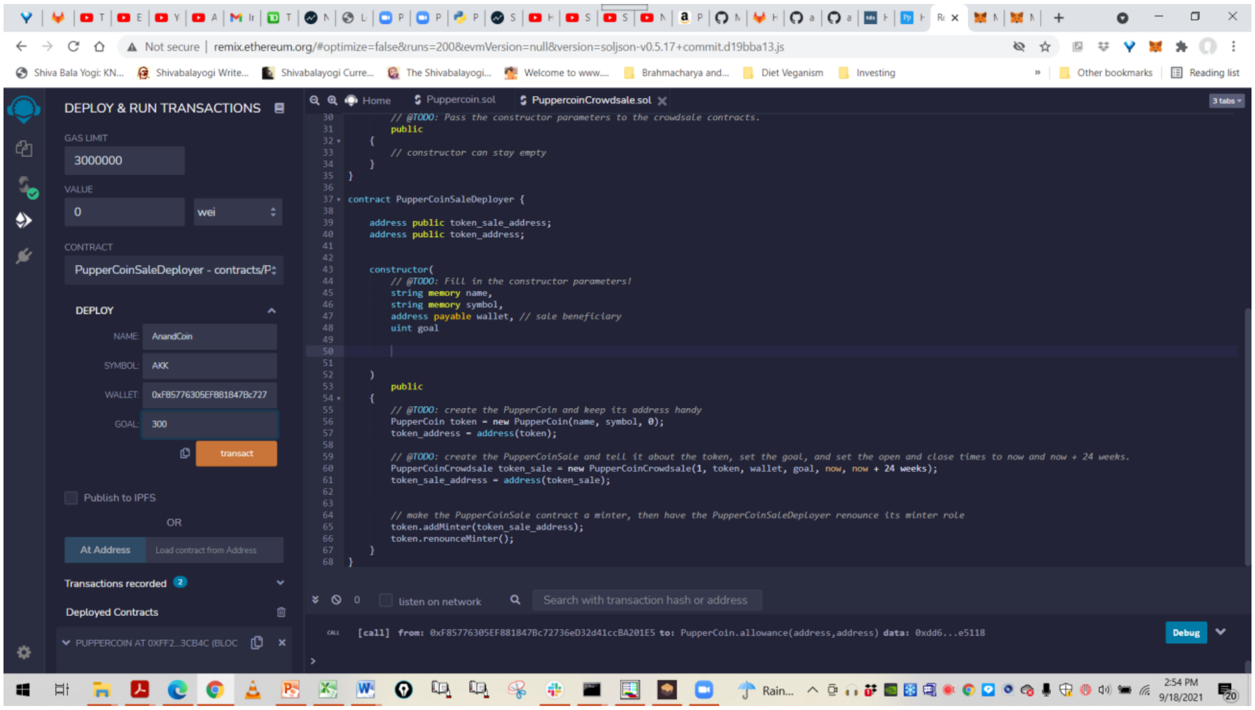
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js

allowance
owner: 0xF85776305EF881847Bc727
spender: 0x63F2C6D3dfF542474289Ed
0: uint256: 0

```
        //uint cap

    )

    Crowdsale(rate, wallet, token)
    TimedCrowdsale(now, now + 24 weeks)
    CappedCrowdsale(goal)
    RefundableCrowdsale(goal)
        // @TODO: Pass the constructor parameters to the crowdsale contracts.
        public
    {
        // constructor can stay empty
    }
}

contract PupperCoinSaleDeployer {

    address public token_sale_address;
    address public token_address;


    constructor(
        // @TODO: Fill in the constructor parameters!
        string memory name,
        string memory symbol,
        address payable wallet, // sale beneficiary
        uint goal



    )
        public
    {
        // @TODO: create the PupperCoin and keep its address handy
        PupperCoin token = new PupperCoin(name, symbol, 0);
        token_address = address(token);

        // @TODO: create the PupperCoinSale and tell it about the token, set the goal, and set the open and close times to now and now + 24 weeks.
        PupperCoinCrowdsale token_sale = new PupperCoinCrowdsale(1, token, wallet, goal, now, now + 24 weeks);
```

```
[call] from: 0xF85776305EF881847Bc72736eD32d41ccBA201E5 to: PupperCoin.allowance(address,address) data: 0xdd6...e5118
```

2:51 PM
9/18/2021



DEPLOY & RUN TRANSACTIONS

GAS LIMIT 3000000

VALUE 0 wei

CONTRACT PupperCoinSaleDeployer - contracts/P

DEPLOY

NAME: AnandCoin

SYMBOL: AKK

WALLET: 0xF85776305EF881847Bc727

GOAL: 300

transact

□ Publish to IPFS

OR

At Address | Load contract from Address

Transactions recorded 2

Deployed Contracts

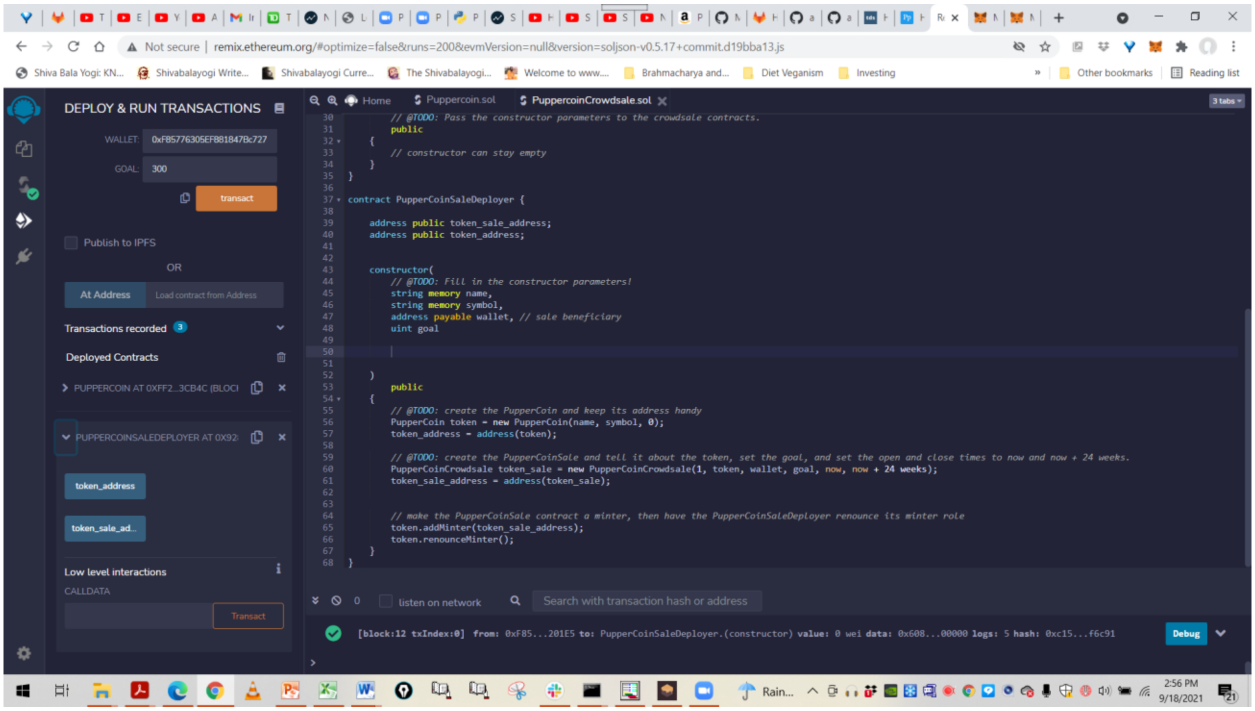
PUPPERCOIN AT 0XFF2...3CB4C (BLOC

```
        // @TODO: Pass the constructor parameters to the crowdsale contracts.
        public
    {
        // constructor can stay empty
    }
}

contract PupperCoinSaleDeployer {

    address public token_sale_address;
    address public token_address;


    constructor(
        // @TODO: Fill in the constructor parameters!
        string memory name,
        string memory symbol,
        address payable wallet, // sale beneficiary
        uint goal



    )
        public
    {
        // @TODO: create the PupperCoin and keep its address handy
        PupperCoin token = new PupperCoin(name, symbol, 0);
        token_address = address(token);

        // @TODO: create the PupperCoinSale and tell it about the token, set the goal, and set the open and close times to now and now + 24 weeks.
        PupperCoinCrowdsale token_sale = new PupperCoinCrowdsale(1, token, wallet, goal, now, now + 24 weeks);
        token_sale_address = address(token_sale);


        // make the PupperCoinSale contract a minter, then have the PupperCoinSaleDeployer renounce its minter role
        token.addMinter(token_sale_address);
        token.renounceMinter();
    }
}
```

0 □ listen on network | Search with transaction hash or address

```
CALL [call] from: 0xF85776305EF881847Bc72736eD32d41ccBA201E5 to: PupperCoin.allowance(address,address) data: 0xdd6...e5118
```

Debug


Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Home
Puppercoin.sol
PuppercoinCrowdsale.sol
3 tabs
Rain...
2:56 PM
9/18/2021

DEPLOY & RUN TRANSACTIONS

WALLET: 0xF85776305EF881847Bc727

GOAL: 300

transact

□ Publish to IPFS

OR

At Address | Load contract from Address

Transactions recorded 3

Deployed Contracts

PUPPERCOIN AT 0XFF2...3CB4C (BLOC

PUPPERCOINSALEDEPLOYER AT 0X92

token_address

token_sale_ad...

Low level interactions

CALLDATA

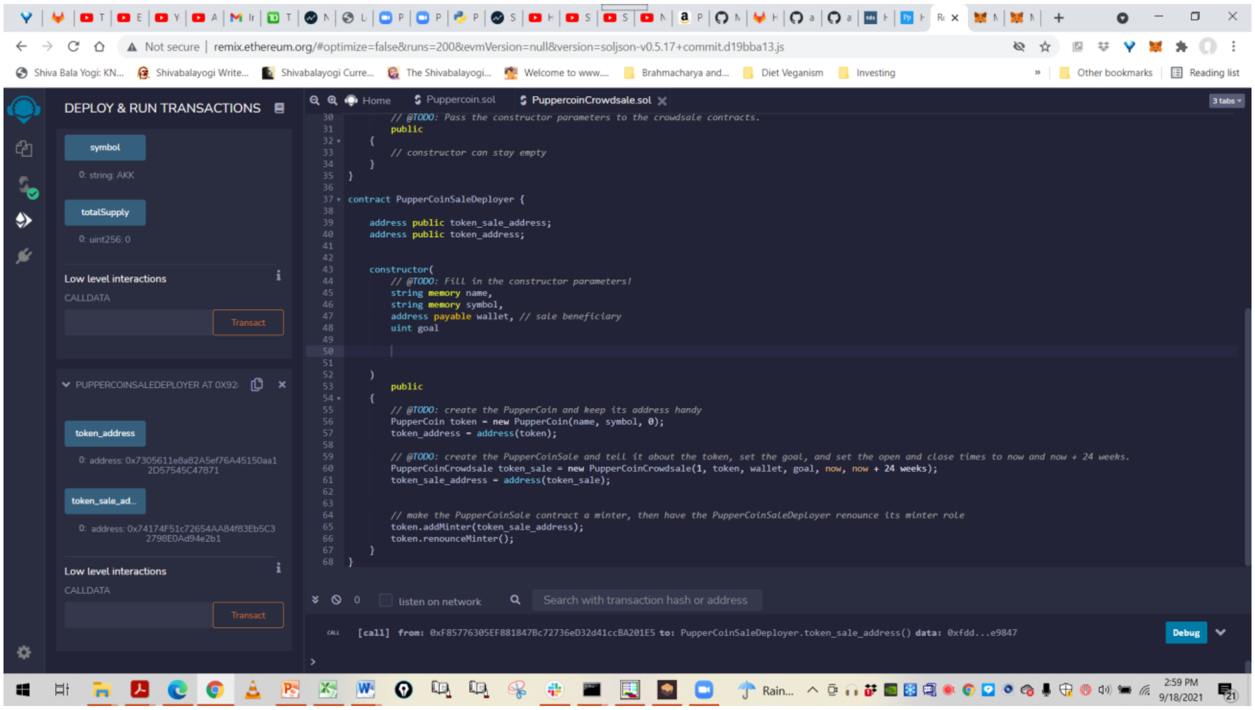
Transact

```
        // @TODO: Pass the constructor parameters to the crowdsale contracts.
        public
    {
        // constructor can stay empty
    }
}

contract PupperCoinSaleDeployer {

    address public token_sale_address;
    address public token_address;


    constructor(
        // @TODO: Fill in the constructor parameters!
        string memory name,
        string memory symbol,
        address payable wallet, // sale beneficiary
        uint goal



    )
        public
    {
        // @TODO: create the PupperCoin and keep its address handy
        PupperCoin token = new PupperCoin(name, symbol, 0);
        token_address = address(token);

        // @TODO: create the PupperCoinSale and tell it about the token, set the goal, and set the open and close times to now and now + 24 weeks.
        PupperCoinCrowdsale token_sale = new PupperCoinCrowdsale(1, token, wallet, goal, now, now + 24 weeks);
        token_sale_address = address(token_sale);


        // make the PupperCoinSale contract a minter, then have the PupperCoinSaleDeployer renounce its minter role
        token.addMinter(token_sale_address);
        token.renounceMinter();
    }
}
```

0 □ listen on network | Search with transaction hash or address

```
✓ [block:12 txIndex:0] from: 0xF85...201E5 to: PupperCoinSaleDeployer.(constructor) value: 0 wei data: 0x608...00000 logs: 5 hash: 0xc15...f6c91
```

Debug



Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Home
Puppercoin.sol
PuppercoinCrowdsale.sol
DEPLOY & RUN TRANSACTIONS

symbol

0: string: AKK

totalSupply

0: uint256: 0

Low level interactions

CALLDATA

Transact

PUPPERCOINSALEDEPLOYER AT 0X92

token_address

0: address: 0x7305611e8a82A5ef76A45150aa12D57545C47871

token_sale_ad...

0: address: 0x74174F51c72654AA84f83Eb5C32798E0Ad94e2b1

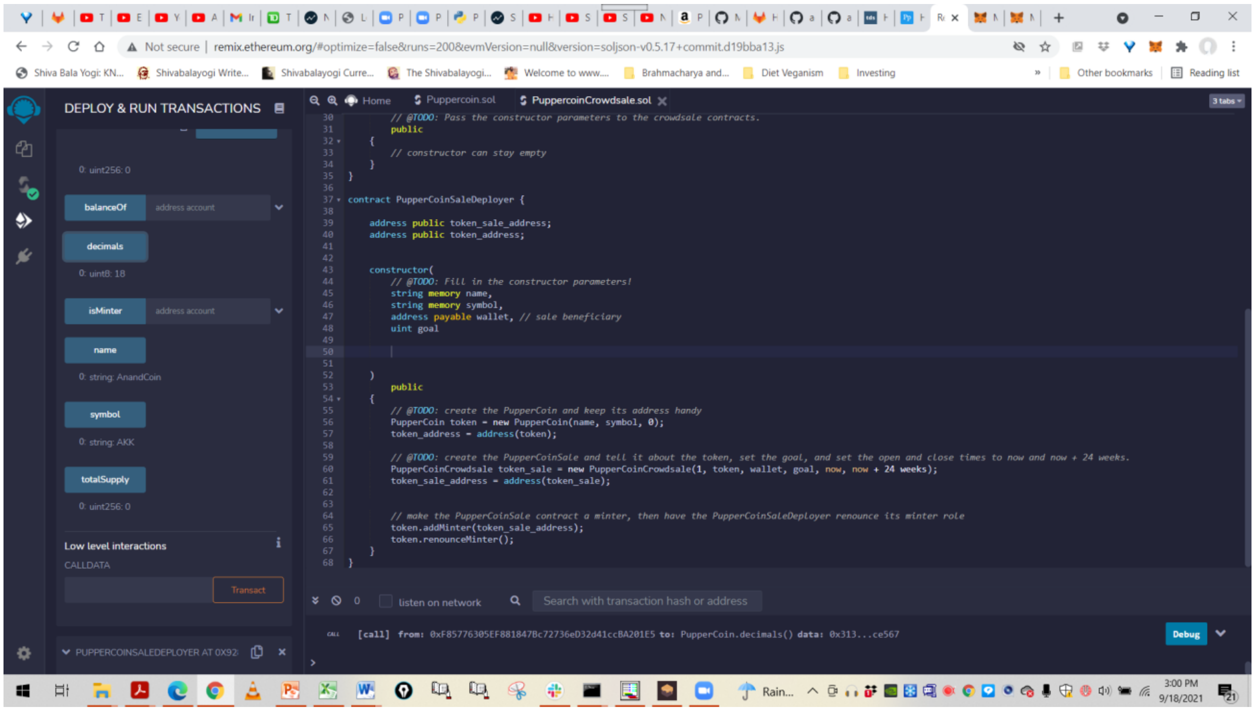
Low level interactions

CALLDATA

Transact

```
        // @TODO: Pass the constructor parameters to the crowdsale contracts.
        public
    {
        // constructor can stay empty
    }
}

contract PupperCoinSaleDeployer {

    address public token_sale_address;
    address public token_address;


    constructor(
        // @TODO: Fill in the constructor parameters!
        string memory name,
        string memory symbol,
        address payable wallet, // sale beneficiary
        uint goal

        


    )
        public
    {
        // @TODO: create the PupperCoin and keep its address handy
        PupperCoin token = new PupperCoin(name, symbol, 0);
        token_address = address(token);

        // @TODO: create the PupperCoinSale and tell it about the token, set the goal, and set the open and close times to now and now + 24 weeks.
        PupperCoinCrowdsale token_sale = new PupperCoinCrowdsale(1, token, wallet, goal, now, now + 24 weeks);
        token_sale_address = address(token_sale);


        // make the PupperCoinSale contract a minter, then have the PupperCoinSaleDeployer renounce its minter role
        token.addMinter(token_sale_address);
        token.renounceMinter();
    }
}
```

```
CALL [call] from: 0xF85776305EF881847Bc72736eD32d41ccBA201E5 to: PupperCoinSaleDeployer.token_sale_address() data: 0xfdd...e9847
```



Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Home
Puppercoin.sol
PuppercoinCrowdsale.sol
DEPLOY & RUN TRANSACTIONS

0: uint256: 0

balanceOf — address account

decimals

0: uint8: 18

isMinter — address account

name

0: string: AnandCoin

symbol

0: string: AKK

totalSupply

0: uint256: 0

Low level interactions

CALLDATA

Transact

PUPPERCOINSALEDEPLOYER AT 0X92

```
        // @TODO: Pass the constructor parameters to the crowdsale contracts.
        public
    {
        // constructor can stay empty
    }
}

contract PupperCoinSaleDeployer {

    address public token_sale_address;
    address public token_address;


    constructor(
        // @TODO: Fill in the constructor parameters!
        string memory name,
        string memory symbol,
        address payable wallet, // sale beneficiary
        uint goal

        


    )
        public
    {
        // @TODO: create the PupperCoin and keep its address handy
        PupperCoin token = new PupperCoin(name, symbol, 0);
        token_address = address(token);

        // @TODO: create the PupperCoinSale and tell it about the token, set the goal, and set the open and close times to now and now + 24 weeks.
        PupperCoinCrowdsale token_sale = new PupperCoinCrowdsale(1, token, wallet, goal, now, now + 24 weeks);
        token_sale_address = address(token_sale);


        // make the PupperCoinSale contract a minter, then have the PupperCoinSaleDeployer renounce its minter role
        token.addMinter(token_sale_address);
        token.renounceMinter();
    }
}
```

```
CALL [call] from: 0xF85776305EF881847Bc72736eD32d41ccBA201E5 to: PupperCoin.decimals() data: 0x313...ce567
```

At this point I was not sure how to get the buy tokens transaction done. So I will submit this for the moment and in future work I will add the rest of it.