

Profit Splitter Smart Contracts

In this activity I create several ProfitSplitter smart contracts with Solidity to accomplish the following across 3 levels of difficulty:

- **Level One** is an AssociateProfitSplitter contract. This will accept Ether into the contract and divide the Ether evenly among the associate level employees. This will allow the Human Resources department to pay employees quickly and efficiently.
- **Level Two** is a TieredProfitSplitter that will distribute different percentages of incoming Ether to employees at different tiers/levels. For example, the CEO gets paid 60%, CTO 25%, and Bob gets 15%.
- **Level Three** is a DeferredEquityPlan that models traditional company stock plans. This contract will automatically manage 1000 shares with an annual distribution of 250 over 4 years for a single employee.

Here is a snapshot of my Ganache workspace before transactions are conducted:

Ganache						
ACCOUNTS	BLOCKS	TRANSACTIONS	CONTRACTS	EVENTS	LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 35	GAS PRICE 2000000000	GAS LIMIT 6721975	HARDFORK PETERSBURG	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:8545	MINING STATUS AUTOMINING
WORKSPACE FINTECH	SWITCH	⚙️				
MNEMONIC ⓘ noble tumble scheme system attack glove soap glare elbow summer tooth robust						
HD PATH m/44'/60'/0'/0/account_index						
ADDRESS 0xf8eF4E21f1501EA4cCfE1110D5541545593Aea4D	BALANCE 49.62 ETH	TX COUNT 35	INDEX 0	🔑		
ADDRESS 0x2997608c322C755b06316959ca9909238A94f781	BALANCE 106.67 ETH	TX COUNT 0	INDEX 1	🔑		
ADDRESS 0xd358861B9DDba6e0D271c96F81225f30Eb7FF6C5	BALANCE 106.67 ETH	TX COUNT 0	INDEX 2	🔑		
ADDRESS 0xDEBf26f27BC532acB60Fb5Fe4C9d03b032c57511	BALANCE 106.67 ETH	TX COUNT 0	INDEX 3	🔑		
ADDRESS 0xA01E53Ce56c045e7d52aFd51c2F94cE2070CbcCb	BALANCE 100.00 ETH	TX COUNT 0	INDEX 4	🔑		
ADDRESS 0xDf4839697eF38b708B8cBD4EA4DCEc3BAAb77aAc	BALANCE 100.00 ETH	TX COUNT 0	INDEX 5	🔑		
ADDRESS 0x56C30753806F009f95bac0200cA92C11844Afe05	BALANCE 100.00 ETH	TX COUNT 0	INDEX 6	🔑		

Level One: The AssociateProfitSplitter Contract

```
pragma solidity ^0.5.0;

// lvl 1: equal split
contract AssociateProfitSplitter {
    address payable employee_one;
    address payable employee_two;
    address payable employee_three;

    constructor(address payable _one, address payable _two, address payable _three) public {
        employee_one = _one;
        employee_two = _two;
        employee_three = _three;
    }

    function balance() public view returns(uint) {
        return address(this).balance;
    }

    function deposit() public payable {
        uint amount = msg.value / 3;

        employee_one.transfer(amount);
        employee_two.transfer(amount);
        employee_three.transfer(amount);

        msg.sender.transfer(msg.value - amount * 3);
    }

    function() external payable {
        deposit();
    }
}
```

The code accepts Ether into the contract, divides the Ether evenly among the three recipient addresses, transfers the calculated amount to the three recipient addresses, and then transfers the remainder back to the sender address.

After successfully compiling the code, we switch the ENVIRONMENT to "Injected Web3" in the dropdown menu in Remix. We paste the three recipient addresses within the DEPLOY area and select "transact" (a value of 0 wei) to deploy our contract to the local network:



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT



Injected Web3



Custom (5777) network

ACCOUNT



0xf8e...Aea4D (49.6160132)



GAS LIMIT



3000000

VALUE

0

wei



CONTRACT

AssociateProfitSplitter - browser/As



DEPLOY

_ONE: 0x2997608c322C755b063169

TWO: 0xd358861B9DDba6e0D271c6

_THREE: 0xDEBf26f27BC532acB60Fb5f



transact

Publish to IPFS

OR

At Address

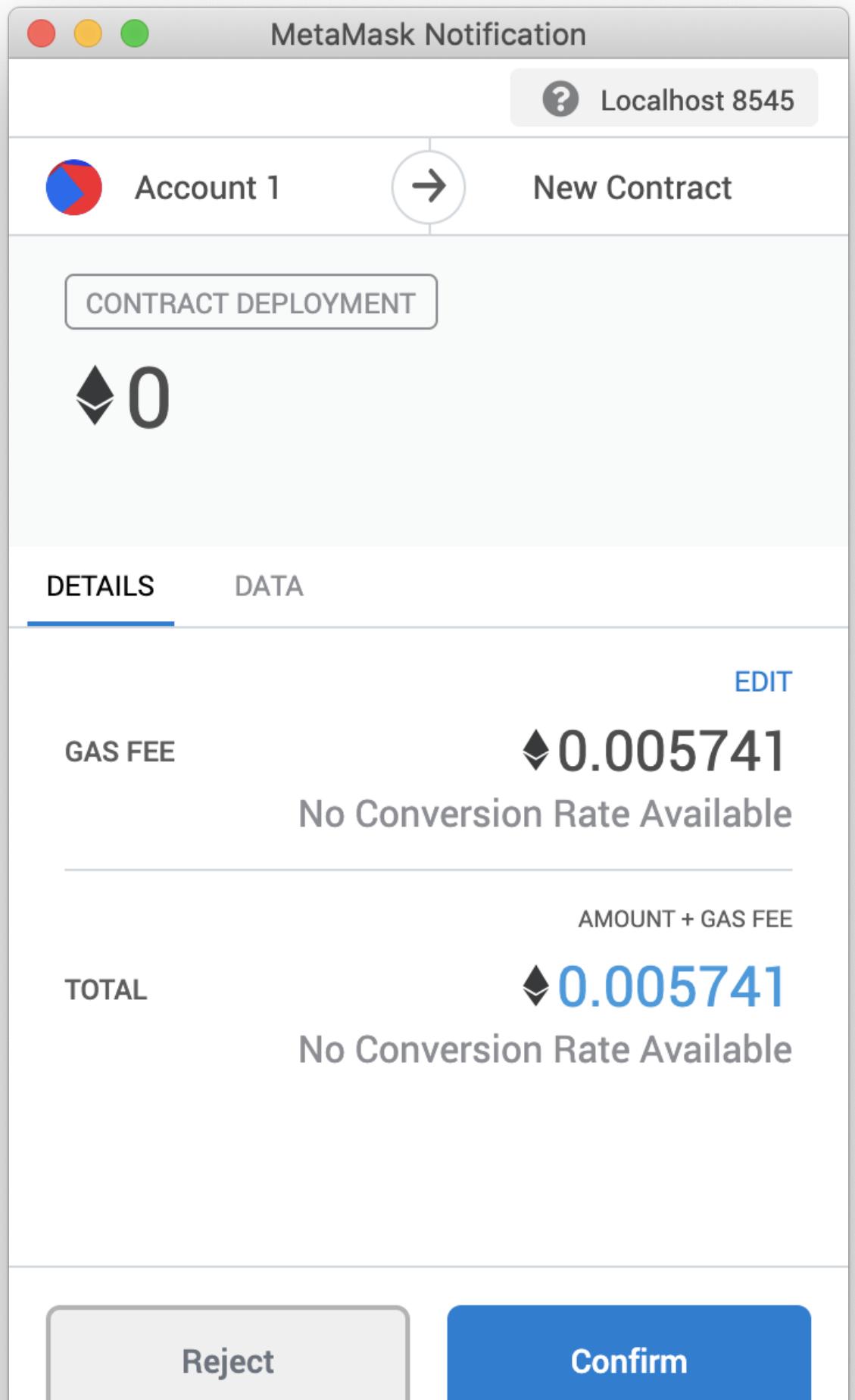
Load contract from Address

Transactions recorded 1

Deployed Contracts



MetaMask pops up and we "Confirm" the contract deployment:



Now we can click an arrow beside "ASSOCIATEPROFITSPLITTER" in Remix to see our options to "deposit" or check "balance." We enter a VALUE of 10 ether to deposit:



DEPLOY & RUN TRANSACTIONS



ENVIRONMENT

Injected Web3  

Custom (5777) network



ACCOUNT

0xf8e...Aea4D (49.6102722)   



GAS LIMIT

3000000



VALUE

10

ether



CONTRACT

AssociateProfitSplitter - browser/As  

DEPLOY

_ONE: 0x2997608c322C755b063169



_TWO: 0xd358861B9DDba6e0D271c5

_THREE: 0xDEBf26f27BC532acB60Fb5F



transact

Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 3

Deployed Contracts



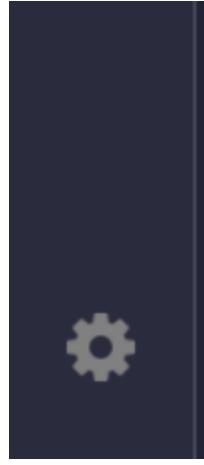
▼ ASSOCIATEPROFITSPLITTER AT 0X9BC...A9116 (BLOCK 116)

deposit

balance

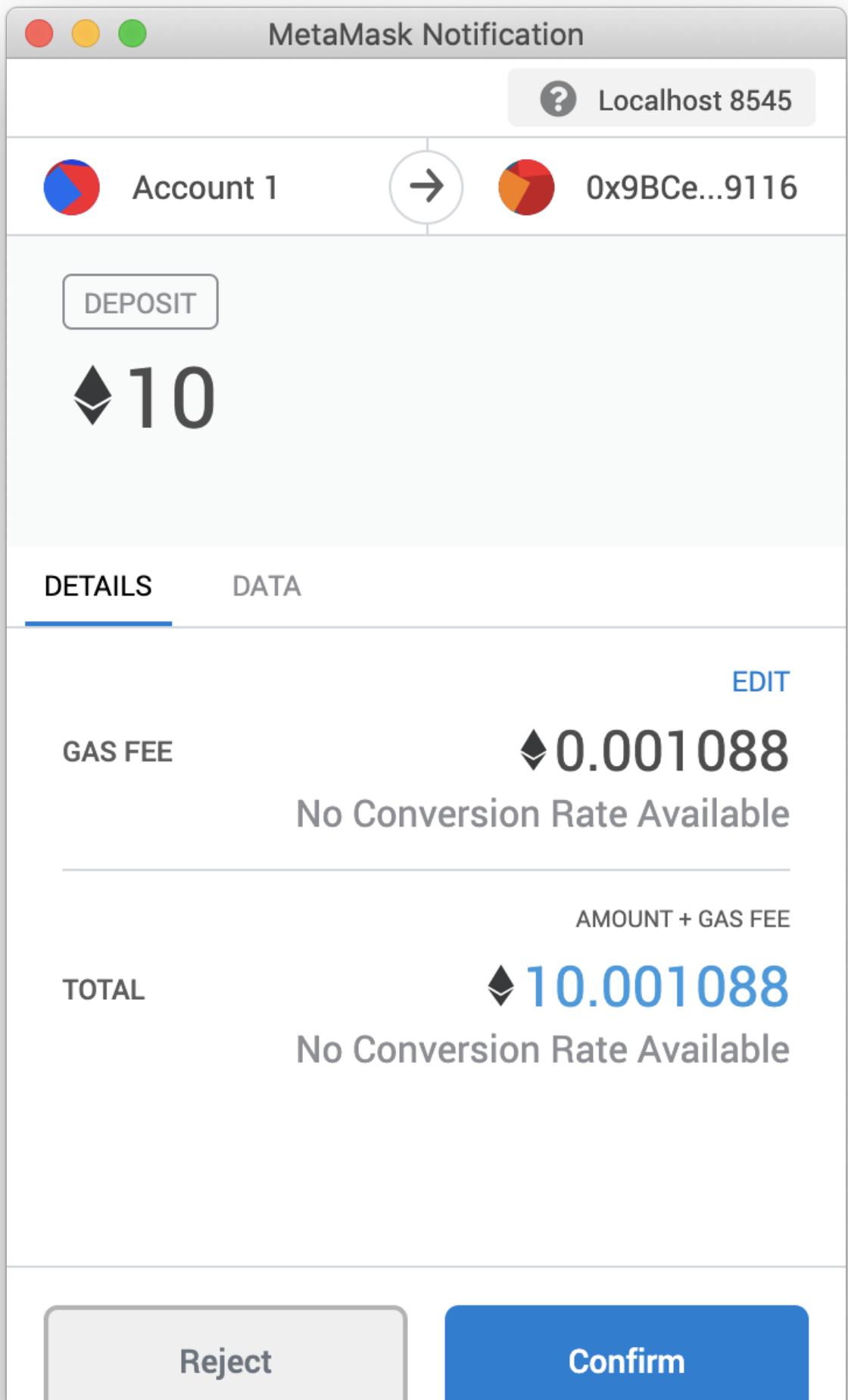
Low level interactions



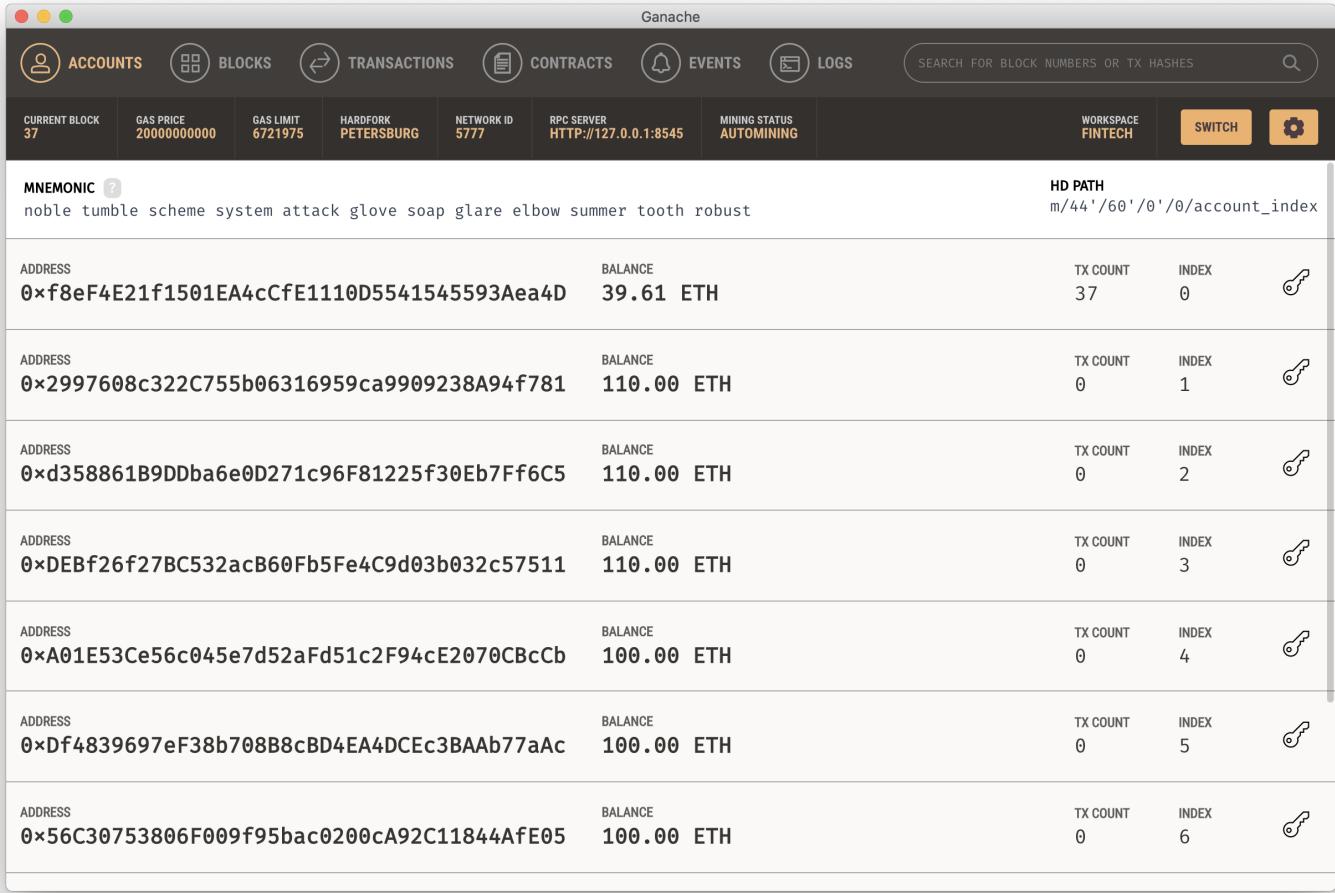


CALLDATA

Transact



We can see in Ganache that the deposit worked!



The screenshot shows the Ganache interface with the following details:

MNEMONIC	HD PATH			
noble tumble scheme system attack glove soap glare elbow summer tooth robust	m/44'/60'/0'/0/account_index			
ADDRESS	BALANCE	TX COUNT	INDEX	
0xf8eF4E21f1501EA4cCfE1110D5541545593Aea4D	39.61 ETH	37	0	🔗
ADDRESS	BALANCE	TX COUNT	INDEX	
0x2997608c322C755b06316959ca9909238A94f781	110.00 ETH	0	1	🔗
ADDRESS	BALANCE	TX COUNT	INDEX	
0xd358861B9DDba6e0D271c96F81225f30Eb7Ff6C5	110.00 ETH	0	2	🔗
ADDRESS	BALANCE	TX COUNT	INDEX	
0xDEBF26f27BC532acB60Fb5Fe4C9d03b032c57511	110.00 ETH	0	3	🔗
ADDRESS	BALANCE	TX COUNT	INDEX	
0xA01E53Ce56c045e7d52aFd51c2F94cE2070CbcCb	100.00 ETH	0	4	🔗
ADDRESS	BALANCE	TX COUNT	INDEX	
0xDf4839697eF38b708B8cBD4EA4DCEc3BAAb77aAc	100.00 ETH	0	5	🔗
ADDRESS	BALANCE	TX COUNT	INDEX	
0x56C30753806F009f95bac0200cA92C11844AFE05	100.00 ETH	0	6	🔗

Now we can point our MetaMask to the Kovan Test Network! We first check our balance in MetaMask on the Kovan Test Network:



Kovan Test Network



Connected

Account 1
0xf8eF...ea4D



0.9733 ETH

BUY

 **SEND**

Assets

Activity



Send ARCDMW

Sep 12 · To: 0x742c...263a

-1e-17

ARCDMW



Send ARCDMW

Sep 12 · To: 0xa9c8...09a7

-1e-17

ARCDMW

We successfully compile our code. Then we change the ENVIRONMENT to Injected Web3 and populate the three recipient addresses under DEPLOY:



DEPLOY & RUN TRANSACTIONS



ENVIRONMENT



Injected Web3



Kovan (42) network



ACCOUNT



0xf8e...Aea4D (0.97327820)



GAS LIMIT



3000000

VALUE

0

wei



CONTRACT

AssociateProfitSplitter - browser/As



DEPLOY



_ONE: 0x2997608c322C755b063169

TWO: 0xd358861B8DDba6e0D271c5

_TWO: 0x6558881B9DDba0e0D271C2

_THREE: 0xDEBf26f27BC532acB60Fb5f



transact



Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 0

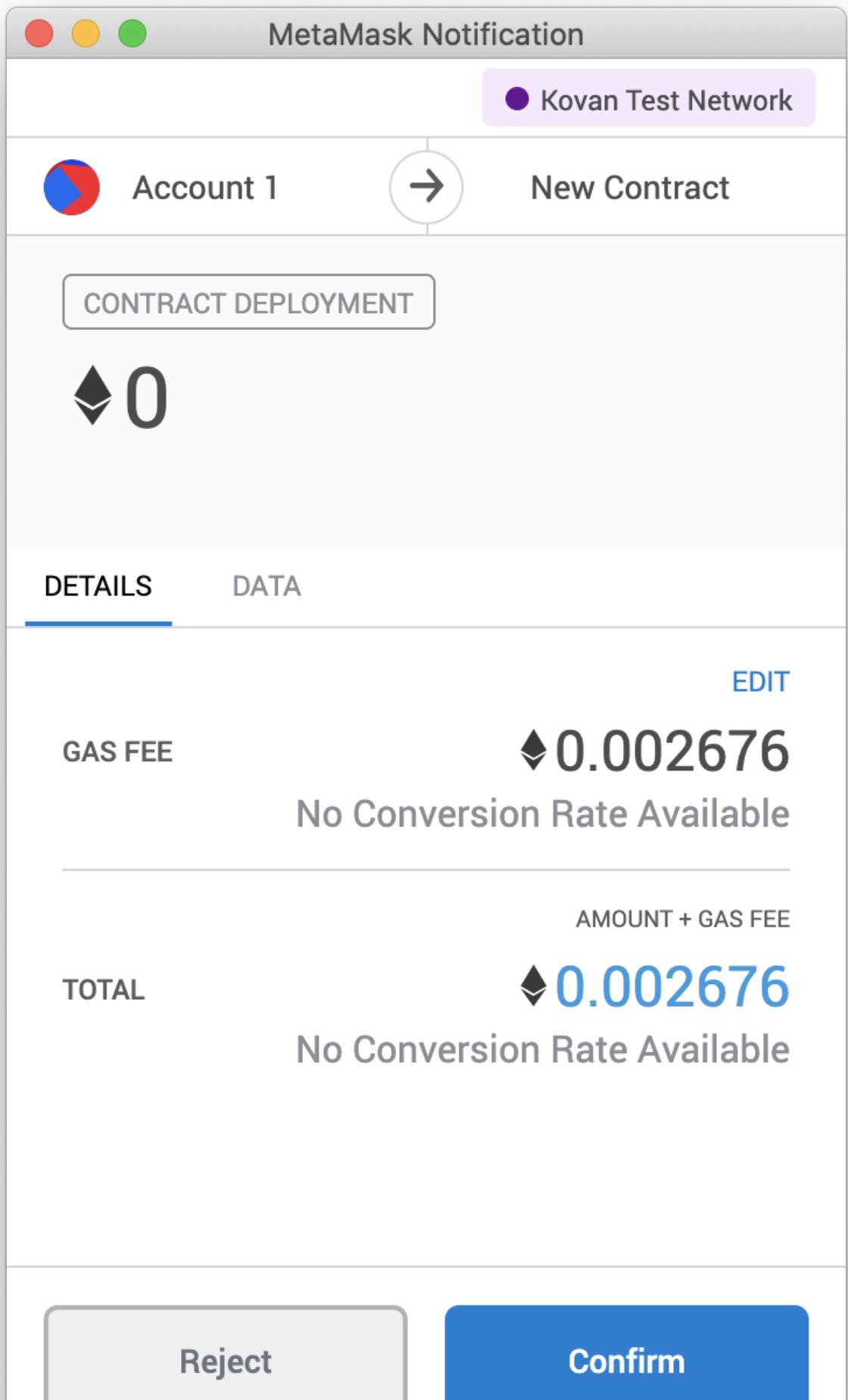


Deployed Contracts



Currently you have no contract instances
to interact with.

We confirm in MetMask:



This generates our option buttons to "deposit" or check "balance" in the lower left. We choose to "deposit" a VALUE of 0.1 ether:



DEPLOY & RUN TRANSACTIONS



ENVIRONMENT

Injected Web3



Kovan (42) network



ACCOUNT

0xf8e...Aea4D (0.97060194)



GAS LIMIT

3000000



VALUE

.1

ether



CONTRACT

AssociateProfitSplitter - browser/As



DEPLOY



_ONE: 0x2997608c322C755b063169

TWO: 0xd358861B9DDba6e0D271c6

_THREE: 0xDEBf26f27BC532acB60Fb5f



transact



Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 1



Deployed Contracts



▼ ASSOCIATEPROFITSPLITTER AT 0xD1E...F31DD (BLOCK 1)

deposit

balance

Low level interactions

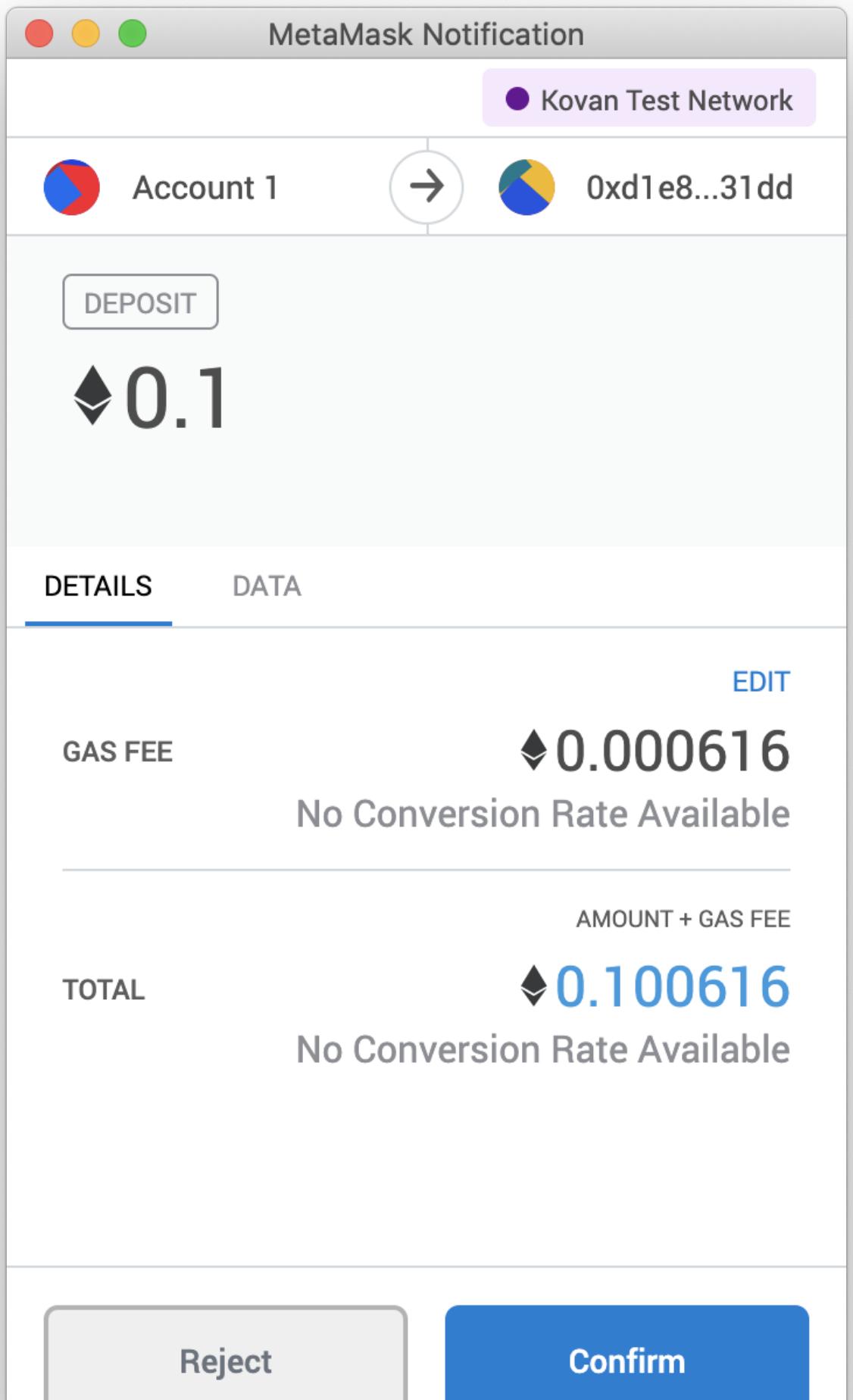


CALldata



Transact

We confirm our deposit in MetaMask:



We can see that the deposit worked and our account balance has decreased:



Kovan Test Network



Connected

Account 1

0xf8eF...ea4D



0.87 ETH

BUY

SEND

Assets

Activity



Deposit

Sep 12 · remix.ethereum.org

-0.1 ETH

-0.1 ETH



Contract Deployment

Sep 12 · remix.ethereum.org

-0 ETH

-0 ETH

We double-check etherscan to determine that the deposit was successfully routed into the three recipient accounts, as well as the remainder of 1 wei that goes back into the sender account:

 **Etherscan**

All Filters Search by Address / Txn Hash / Block /

Kovan Testnet Network

Transaction Details

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

② Transaction Hash:	0x8307803eb2782cce75891324f6c7aee757663019980d1ce527e10a94c409336a <input type="button" value="🔗"/>
② Status:	Success
② Block:	20875183 <small>32 Block Confirmations</small>
② Timestamp:	② 2 mins ago (Sep-12-2020 09:18:08 PM +UTC)
② From:	0xf8ef4e21f1501ea4ccfe1110d5541545593aea4d <input type="button" value="🔗"/>
② To:	② Contract 0xd1e8b791c957ad0f008611dc1f7e4339ca2f31dd ✓ <input type="button" value="🔗"/> ↳ TRANSFER 0.0333333333333333 Ether From 0xd1e8b791c957ad0f008611... To → 0x2997608c322c755b063169... ↳ TRANSFER 0.0333333333333333 Ether From 0xd1e8b791c957ad0f008611... To → 0xd358861b9ddba6e0d271c... ↳ TRANSFER 0.0333333333333333 Ether From 0xd1e8b791c957ad0f008611... To → 0xdebfb26f27bc532acb60fb5f... ↳ TRANSFER 1 wei From 0xd1e8b791c957ad0f008611... To → 0xf8ef4e21f1501ea4ccfe1110...
② Value:	0.1 Ether (\$0.00)
② Transaction Fee:	0.000591327 Ether (\$0.000000)

[Click to see More](#)

The testnet address for others to be able to send to is:

0xf8ef4e21f1501EA4cCfE1110D5541545593Aea4D.

Level Two: The TieredProfitSplitter Contract

```

pragma solidity ^0.5.0;

// lvl 2: tiered split
contract TieredProfitSplitter {
    address payable employee_one; // ceo
    address payable employee_two; // cto
    address payable employee_three; // bob

    constructor(address payable _one, address payable _two, address payable _three) public {
        employee_one = _one;
        employee_two = _two;
        employee_three = _three;
    }

    // Should always return 0! Use this to test your `deposit` function's logic
    function balance() public view returns(uint) {
        return address(this).balance;
    }

    function deposit() public payable {
        uint points = msg.value / 100; // Calculates rudimentary percentage by dividing msg.value into
        uint total;
        uint amount;

        // @TODO: Calculate and transfer the distribution percentage
        // Step 1: Set amount to equal `points` * the number of percentage points for this employee
        // Step 2: Add the `amount` to `total` to keep a running total
        // Step 3: Transfer the `amount` to the employee

        amount = points * 60;
        total += amount;
        employee_one.transfer(amount);

        // @TODO: Repeat the previous steps for `employee_two` and `employee_three`

        amount = points * 25;
        total += amount;
        employee_two.transfer(amount);

        amount = points * 15;
        total += amount;
        employee_three.transfer(amount);

        employee_one.transfer(msg.value - total); // ceo gets the remaining wei
    }

    function() external payable {
        deposit();
    }
}

```

The code calculates one percentage of the transferred amount and then splits it among three recipient accounts with different tiers of compensation. It also keeps a running total of the amounts transferred to each employee and then transfers any remainder to the employee with the highest percentage by subtracting the running total from msg.value.

We switch back to our Localhost 8545 network and successfully compile the code, then switch the ENVIRONMENT to Injected Web3. We enter the three recipient addresses and click "transact" with a VALUE of 0 wei:



DEPLOY & RUN TRANSACTIONS



ENVIRONMENT

Injected Web3



Custom (5777) network



ACCOUNT

0xf8e...Aea4D (39.6092289)



GAS LIMIT

3000000



VALUE

0

wei



CONTRACT

TieredProfitSplitter - browser/Tiered



DEPLOY

_ONE: 0x2997608c322C755b063169



_TWO: 0xd358861B9DDba6e0D271c9

_THREE: 0xDEBf26f27BC532acB60Fb5f



transact

Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 0



Deployed Contracts



Currently you have no contract instances
to interact with.

MetaMask pops up and we Confirm:

MetaMask Notification

localhost 8545

Account 1 → New Contract

CONTRACT DEPLOYMENT

0

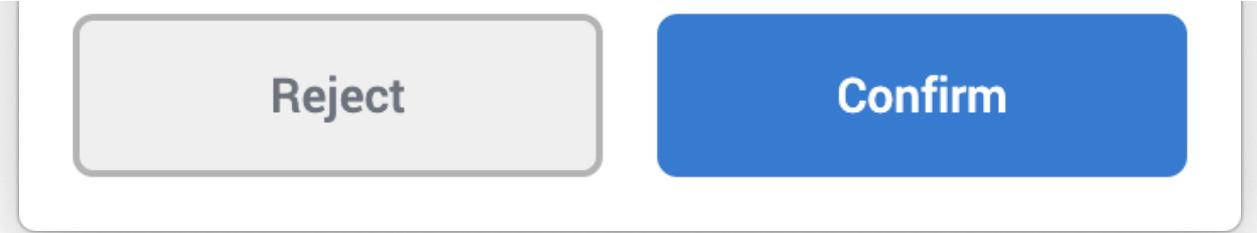
DETAILS DATA

[EDIT](#)

GAS FEE **0.006144**
No Conversion Rate Available

AMOUNT + GAS FEE

0.006144
No Conversion Rate Available



Reject

Confirm

Now we can view the dropdown "deposit" and "balance" buttons in the lower left of Remix. We choose to transfer a VALUE of 10 ether and click "deposit":



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT



Injected Web3



Custom (5777) network



ACCOUNT

0xf8e...Aea4D (39.6030853)



GAS LIMIT



3000000

VALUE

10

ether



CONTRACT

TieredProfitSplitter - browser/Tiered



DEPLOY



_ONE: 0x2997608c322C755b063169

_TWO: 0xd358861B9DDba6e0D271c9

_THREE: 0xDEBf26f27BC532acB60Fb5f



transact

Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 1



Deployed Contracts



▼ TIEREDPROFITSPLITTER AT 0X4E2...16EE8 (BLOCKCHA

deposit

balance

Low level interactions

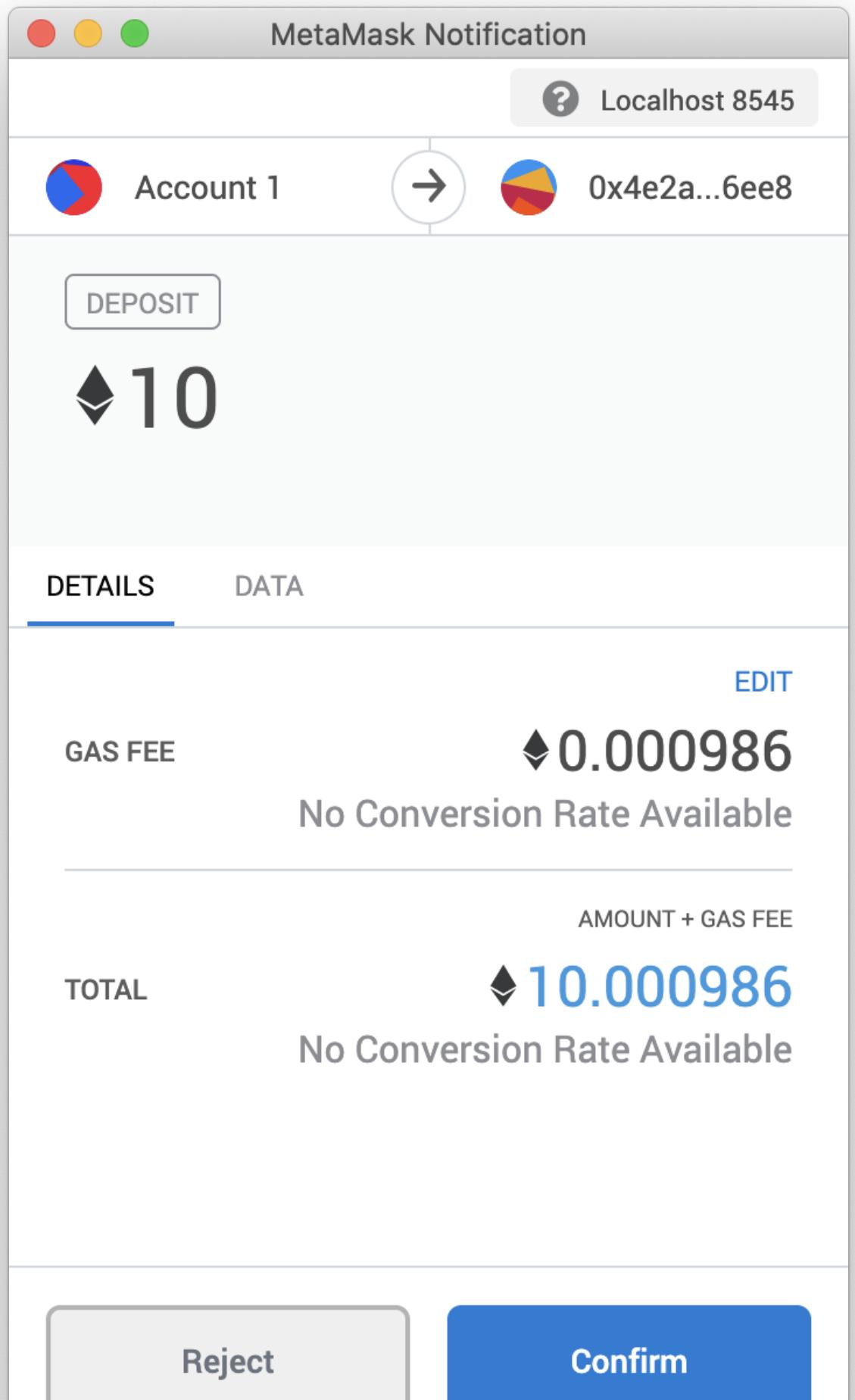


CALldata

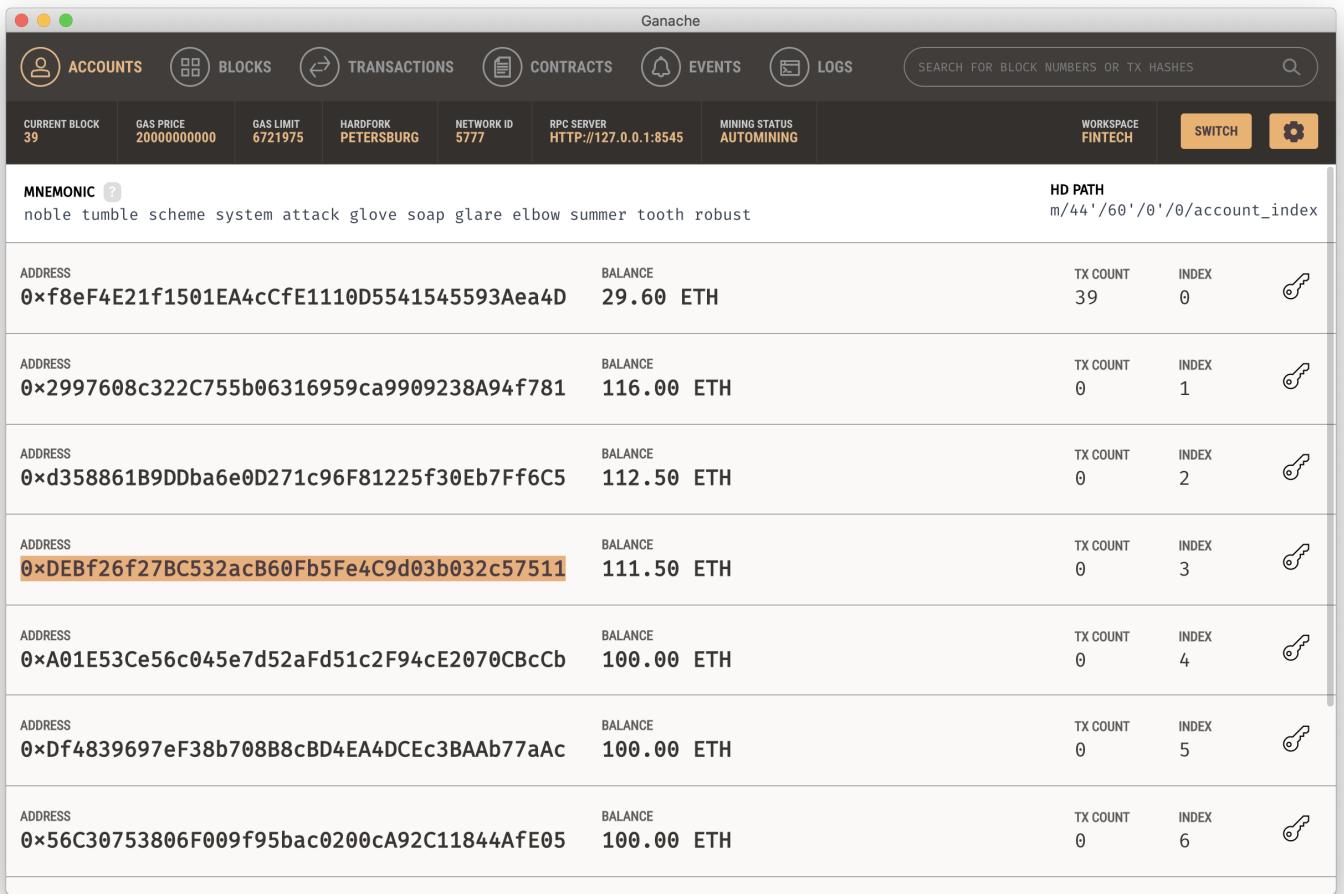


Transact

MetaMask pops up and we confirm the deposit:



We are able to see in Ganache that our deposits were successful and our account balances have updated!



The screenshot shows the Ganache interface with the following details:

MNEMONIC	HD PATH			
?	m/44'/60'/0'/0/account_index			
ADDRESS	BALANCE	TX COUNT	INDEX	
0xf8eF4E21f1501EA4cCfE1110D5541545593Aea4D	29.60 ETH	39	0	🔗
ADDRESS	BALANCE	TX COUNT	INDEX	
0x2997608c322C755b06316959ca9909238A94f781	116.00 ETH	0	1	🔗
ADDRESS	BALANCE	TX COUNT	INDEX	
0xd358861B9DDba6e0D271c96F81225f30Eb7Ff6C5	112.50 ETH	0	2	🔗
ADDRESS	BALANCE	TX COUNT	INDEX	
0xDEBf26f27BC532acB60Fb5Fe4C9d03b032c57511	111.50 ETH	0	3	🔗
ADDRESS	BALANCE	TX COUNT	INDEX	
0xA01E53Ce56c045e7d52aFd51c2F94cE2070CBcCb	100.00 ETH	0	4	🔗
ADDRESS	BALANCE	TX COUNT	INDEX	
0xDf4839697eF38b708B8cBD4EA4DCEc3BAAb77aAc	100.00 ETH	0	5	🔗
ADDRESS	BALANCE	TX COUNT	INDEX	
0x56C30753806F009f95bac0200cA92C11844AFE05	100.00 ETH	0	6	🔗

Now we switch back to the Kovan Test Network in MetaMask:



Kovan Test Network



Connected

Account 1

0xf8eF...ea4D



0.87 ETH

BUY

SEND

Assets

Activity



Deposit

Sep 12 · remix.ethereum.org

-0.1 ETH

-0.1 ETH



Contract Deployment

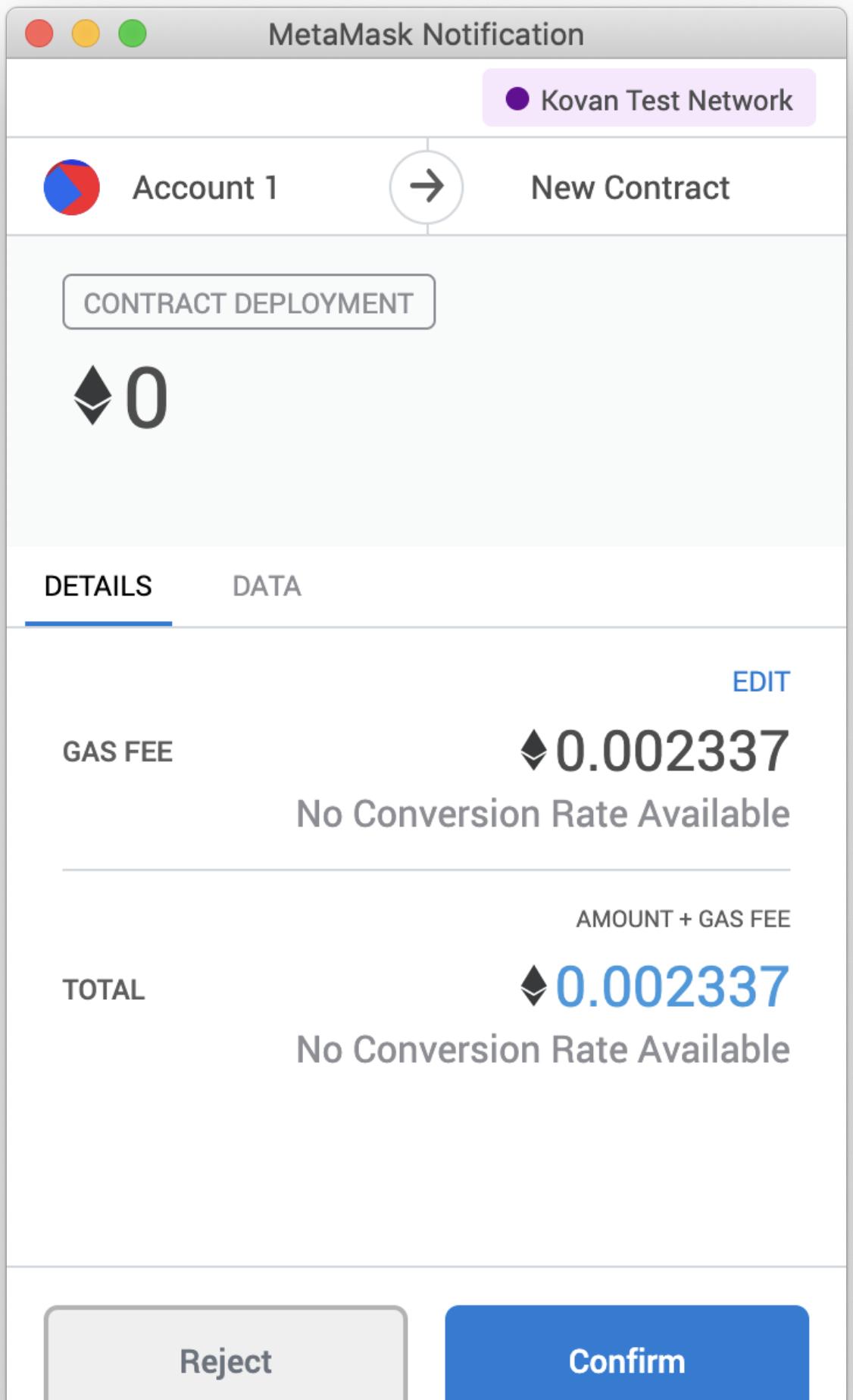
Sep 12 · remix.ethereum.org

-0 ETH

-0 ETH

~~steps for employee two and employee three~~

We successfully compile the code, switch the Remix ENVIRONMENT to Injected Web3, enter the three recipient addresses, and "transact" 0 wei to deploy the contract. We Confirm in MetaMask:



Now we can see "TIEREDPROFITSPLITTER" in the lower-left dropdown which reveals the "deposit" and "balance" buttons. We choose a VALUE of 0.1 ether to deposit:



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT



Injected Web3



Kovan (42) network



ACCOUNT

0xf8e...Aea4D (0.86767346)



GAS LIMIT

3000000

VALUE

.1

ether



CONTRACT

TieredProfitSplitter - browser/Tiered



DEPLOY



_ONE: 0x2997608c322C755b063169

_TWO: 0xd358861B9DDba6e0D271c5

_THREE: 0xDEBf26f27BC532acB60Fb5f



transact

Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 1



Deployed Contracts



▼ TIEREDPROFITSPLITTER AT 0X9D2...4DF3C (BLOCKCH

deposit

balance

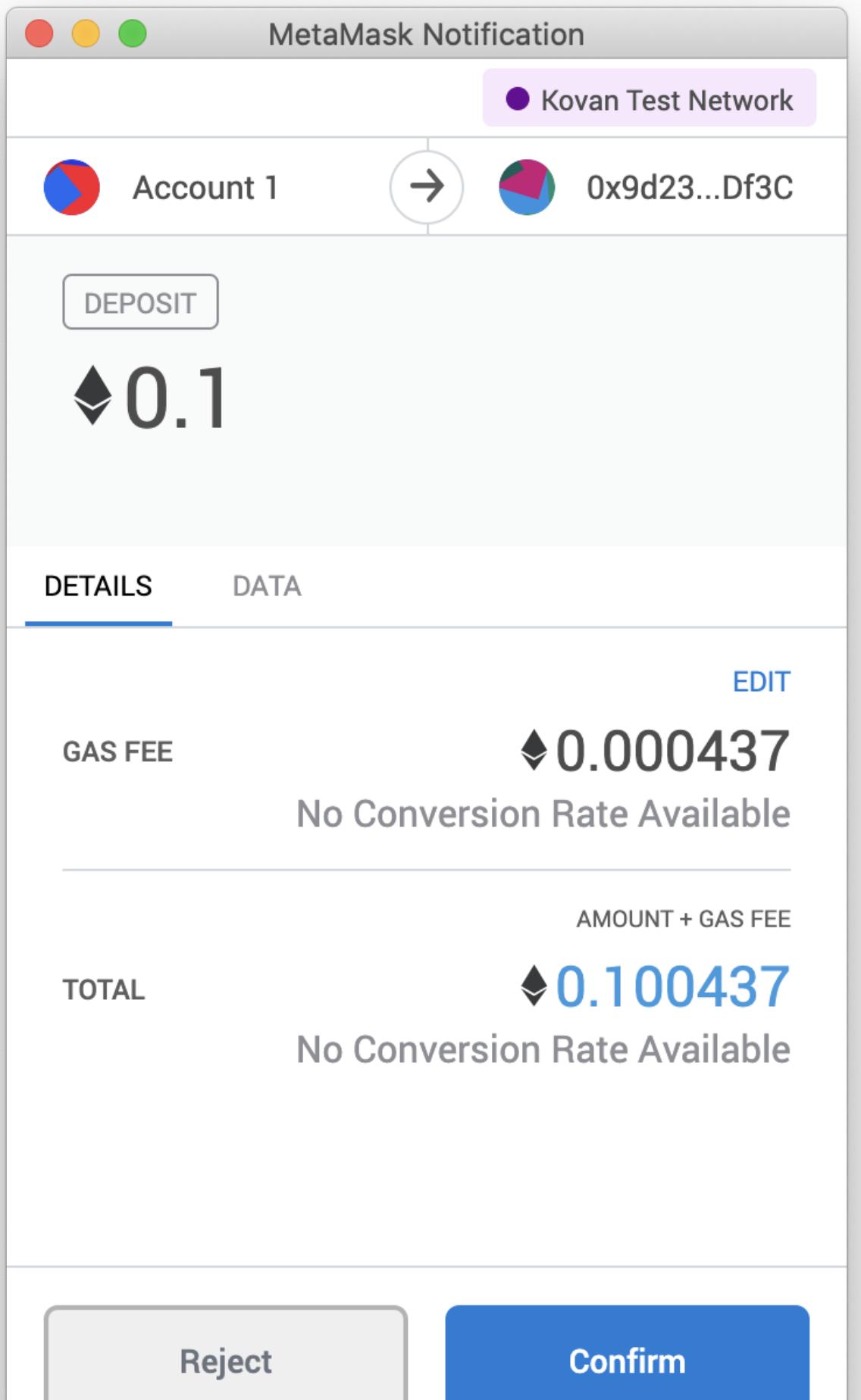
Low level interactions



CALldata

Transact

We Confirm in the MetaMask Notification popup:



We can see the transaction "pending":



Kovan Test Network



Connected

Account 1

0xf8eF...ea4D



0.8677 ETH

BUY

SEND

Assets

Activity

Queue (1)



Deposit



Pending · remix.ethereum.org

-0.1 ETH

-0.1 ETH

Speed Up

Cancel

History

We can use MetaMask to view the transaction details and confirm successful execution:

Deposit



Details

[View on Etherscan](#)

From: 0xf8eF4E21f150... > To: 0x9d232AC8DAFcE...

Transaction

Nonce	11
Amount	0.1 ETH
Gas Limit (Units)	48590
Gas Used (Units)	47986
Gas Price (GWEI)	9
Total	0.100432 ETH

Activity Log



Transaction created with a value of 0.1 ETH at 16:58 on 9/12/2020.

Transaction submitted with gas fee of 437310 GWEI at 16:58 on 9/12/2020.

GWET at 16:59 on 9/12/2020.

Transaction confirmed at 17:00 on 9/12/2020.

 **Etherscan**

Kovan Testnet Network

All Filters ▼ Search by Address / Txn Hash / Block / 🔍

Home Blockchain ▼ Tokens ▼ Misc ▼ Kovan

Transaction Details

[Overview](#) [Internal Txns](#) [State](#) ⋮

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

② Transaction Hash: [0x68f04574e3f1507d5a25dc73f2a38ec4a291dac4e70e49ba930d42303f3c3a9f](#) 📋

② Status: Success

② Block: [20875808](#) 32 Block Confirmations

② Timestamp: [2 mins ago \(Sep-12-2020 09:59:48 PM +UTC\)](#)

② From: [0xf8ef4e21f1501ea4ccfe1110d5541545593aea4d](#) 📋

② To: [Contract 0x9d232ac8dafceabc67edea503dbeeb4a7e64df3c](#) ✓ 📋
└ TRANSFER 0.06 Ether From [0x9d232ac8dafceabc67edea...](#) To → [0x2997608c322c755b063169...](#)
└ TRANSFER 0.025 Ether From [0x9d232ac8dafceabc67edea...](#) To → [0xd358861b9ddba6e0d271c...](#)
└ TRANSFER 0.015 Ether From [0x9d232ac8dafceabc67edea...](#) To → [0xdebff26f27bc532acb60fb5f...](#)

② Value: [0.1 Ether \(\\$0.00\)](#)

② Transaction Fee: [0.000431874 Ether \(\\$0.000000\)](#)

[Click to see More](#) ▼

We now check the "balance" which is 0, as expected:



DEPLOY & RUN TRANSACTIONS



ENVIRONMENT

Injected Web3



Kovan (42) network

ACCOUNT

0xf8e...Aea4D (0.76724159)



GAS LIMIT

3000000



VALUE

0

ether



CONTRACT

TieredProfitSplitter - browser/Tiered



DEPLOY



_ONE: 0x2997608c322C755b063169

TWO: 0xd358861B9DDba6e0D271c9



_THREE: 0xDEBf26f27BC532acB60Fb5f



transact



Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 2



Deployed Contracts



▼ TIEREDPROFITSPLITTER AT 0X9D2...4DF3C (BLOCKCH

deposit

balance

0: uint256: 0

Low level interactions





CALLDATA

Transact

Again, the testnet address for others to be able to send to is:

0xf8eF4E21f1501EA4cCfE1110D5541545593Aea4D.

Level Three: The DeferredEquityPlan Contract

```

pragma solidity ^0.5.0;

// lvl 3: equity plan
contract DeferredEquityPlan {
    uint fakenow = now;
    function fastforward() public {
        fakenow += 100 days;
    }
    address human_resources;

    address payable employee; // bob
    bool active = true; // this employee is active at the start of the contract

    // @TODO: Set the total shares and annual distribution
    uint total_shares = 1000;
    uint annual_distribution = 250;

    uint start_time = fakenow; // permanently store the time this contract was initialized

    // @TODO: Set the `unlock_time` to be 365 days from now
    uint unlock_time = fakenow + 365 days;

    uint public distributed_shares; // starts at 0

    constructor(address payable _employee) public {
        human_resources = msg.sender;
        employee = _employee;
    }

    function distribute() public {
        require(msg.sender == human_resources || msg.sender == employee, "You are not authorized to execute this function");
        require(active == true, "Contract not active.");

        // @TODO: Add "require" statements to enforce that:
        // 1: `unlock_time` is less than or equal to `now`
        require(unlock_time <= fakenow, "Account is locked!");
        // 2: `distributed_shares` is less than the `total_shares`
        require(distributed_shares < total_shares, "Available shares have been exceeded!");

        // @TODO: Add 365 days to the `unlock_time`
        unlock_time += 365 days;

        // @TODO: Calculate the shares distributed by using the function (now - start_time) / 365 days
        // Make sure to include the parenthesis around (now - start_time) to get accurate results!
        distributed_shares = (fakenow - start_time) / 365 days * annual_distribution;

        // double check in case the employee does not cash out until after 5+ years
        if (distributed_shares > 1000) {
            distributed_shares = 1000;
        }
    }
}

```

```
}

// human_resources and the employee can deactivate this contract at-will
function deactivate() public {
    require(msg.sender == human_resources || msg.sender == employee, "You are not authorized to deactivate");
    active = false;
}

// Since we do not need to handle Ether in this contract, revert any Ether sent to the contract directly
function() external payable {
    revert("Do not send Ether to this contract!");
}

}
```

The code works by keeping track of how much time has passed since the `start_time`, and storing and setting amounts that represent the number of distributed shares the employee owns, enforcing the vesting periods automatically. We use a "fakenow" variable that allows us to skip through time for testing purposes.

We switch back to our Localhost 8545 network and successfully compile the code, then switch the ENVIRONMENT to Injected Web3. We enter the recipient address and click "transact" with a VALUE of 0 wei:



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT



Injected Web3  



Custom (5777) network

ACCOUNT



0xf8e...Aea4D (29.6021695)   

GAS LIMIT



3000000

VALUE

0

wei



CONTRACT

DeferredEquityPlan - browser/Defer  

DEPLOY

_EMPLOYEE: 0x2997608c322C755b063169





transact



Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 0

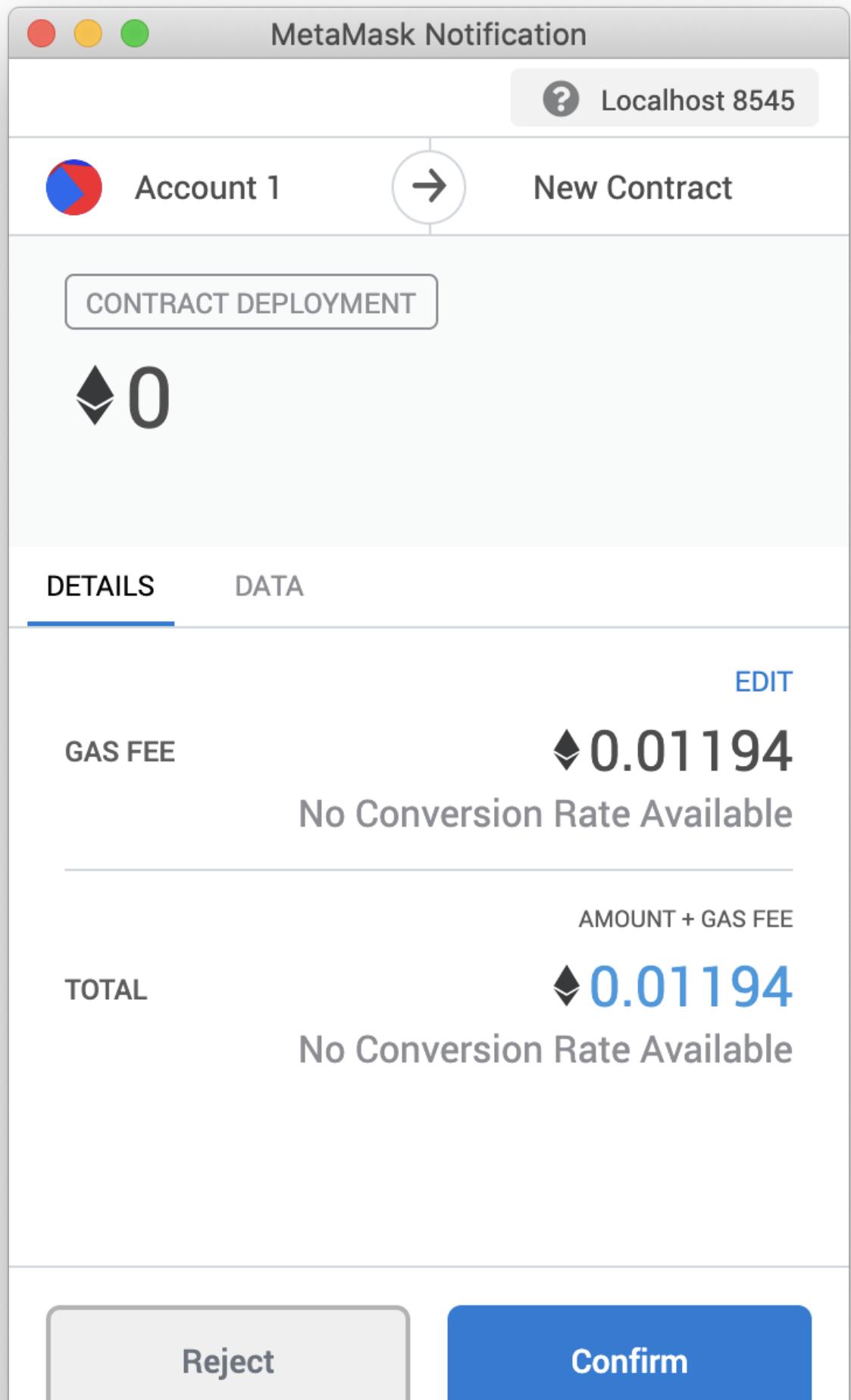


Deployed Contracts



Currently you have no contract instances
to interact with.

We confirm the deployment in the MetaMask popup:



Now we can see the "deactivate," "distribute," "fastforward," and "distributed_shares" buttons in the lower-left dropdown. We choose "distributed_shares" and see that there are 0 distributed shares:



DEPLOY & RUN TRANSACTIONS



ENVIRONMENT

Injected Web3

Custom (5777) network

ACCOUNT

0xf8e...Aea4D (29.5902296)

GAS LIMIT

3000000

VALUE

0

ether



CONTRACT

DeferredEquityPlan - browser/Defer

DEPLOY

_EMPLOYEE: 0x2997608c322C755b063169



transact

Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 1

Deployed Contracts



▼ DEFERREDEQUITYPLAN AT 0XABA...832FD (BLOCKCH

deactivate

distribute

fastforward

distributed_sh...

0: uint256: 0

Low level interactions

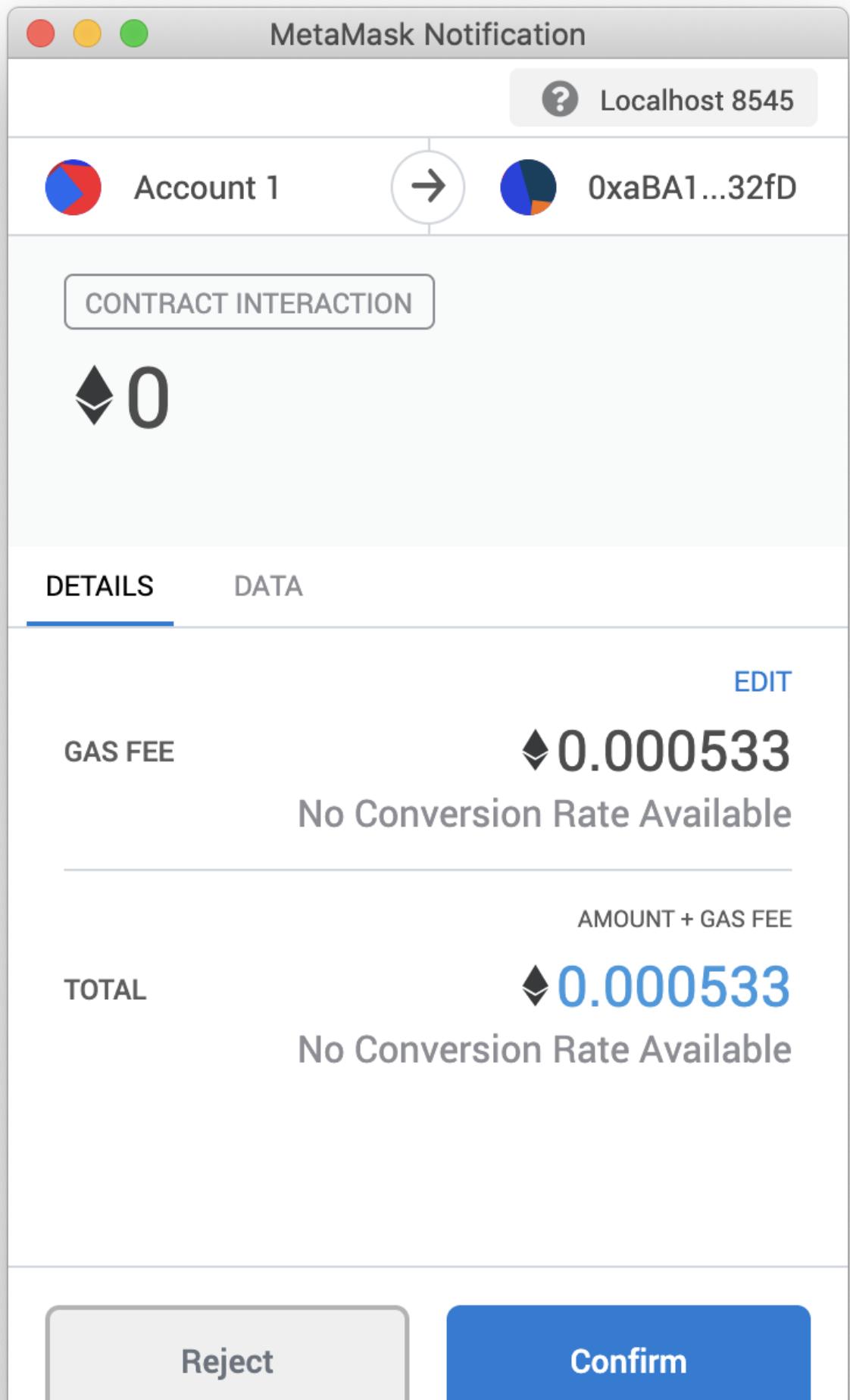
i

CALldata



Transact

We then choose "fastforward" and Confirm in the MetaMask popup:



We still see 0 distributed shares because not enough time has passed, so we click "fastforward" three more times, confirming in the MetaMask popup each time. Now we are able to choose "distribute" and Confirm in the MetaMask popup. Now when we click "distributed_share" we see there are 250 distributed shares, as expected!



DEPLOY & RUN TRANSACTIONS



Injected Web3



Custom (5777) network

ACCOUNT

0xf8e...Aea4D (29.5871171)



GAS LIMIT

3000000

VALUE

0

ether



CONTRACT

DeferredEquityPlan - browser/Defer



DEPLOY



_EMPLOYEE:

0x2997608c322C755b063169



transact



Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 7

Deployed Contracts



▼ DEFERREDEQUITYPLAN AT 0XABA...832FD (BLOCKCH

deactivate

distribute

fastforward

distributed_sh...

0: uint256: 250

Low level interactions



CALLDATA

Transact



We can see these recent transactions in Ganache:

Ganache						SEARCH FOR BLOCK NUMBERS OR TX HASHES	WORKSPACE	SWITCH	⚙️
CURRENT BLOCK 45	GAS PRICE 20000000000	GAS LIMIT 6721975	HARDFORK PETERSBURG	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:8545	MINING STATUS AUTOMINING			
TX HASH 0x01768cf599825bebf874f39e5e50a4de7cab3eea3769d785385ed9d05daaaa64								CONTRACT CALL	
FROM ADDRESS 0xf8eF4E21f1501EA4cCfe1110D5541545593Aea4D					TO CONTRACT ADDRESS 0xaBA14A3eeF276ad89003F30327374E73E33832fd		GAS USED 49029	VALUE 0	
TX HASH 0x8cf1741b8c89b465972e257a8c2e13eadf58e8f0ca05bdd691e1e7a038e440d								CONTRACT CALL	
FROM ADDRESS 0xf8eF4E21f1501EA4cCfe1110D5541545593Aea4D					TO CONTRACT ADDRESS 0xaBA14A3eeF276ad89003F30327374E73E33832fd		GAS USED 26649	VALUE 0	
TX HASH 0x6a0a9bf5542e023441e815d9d101a3ecb8d976330271359d0d0aadabe9f1f6dc								CONTRACT CALL	
FROM ADDRESS 0xf8eF4E21f1501EA4cCfe1110D5541545593Aea4D					TO CONTRACT ADDRESS 0xaBA14A3eeF276ad89003F30327374E73E33832fd		GAS USED 26649	VALUE 0	
TX HASH 0x26a18f6eeb8689b05be0d2fcccaf7886ad2a4e7d72e7d5a6906a22e4357c610f								CONTRACT CALL	
FROM ADDRESS 0xf8eF4E21f1501EA4cCfe1110D5541545593Aea4D					TO CONTRACT ADDRESS 0xaBA14A3eeF276ad89003F30327374E73E33832fd		GAS USED 26649	VALUE 0	
TX HASH 0xfb3f9c1a54333b02166ab83398818965e0477ad02b822af050d81a69f9566d8								CONTRACT CALL	
FROM ADDRESS 0xf8eF4E21f1501EA4cCfe1110D5541545593Aea4D					TO CONTRACT ADDRESS 0xaBA14A3eeF276ad89003F30327374E73E33832fd		GAS USED 26649	VALUE 0	
TX HASH 0xcf03f4757b13d0773fe409ed534b3ad04c7dba92616c7d380fa806445498d76								CONTRACT CREATION	
FROM ADDRESS 0xf8eF4E21f1501EA4cCfe1110D5541545593Aea4D					CREATED CONTRACT ADDRESS 0xaBA14A3eeF276ad89003F30327374E73E33832fd		GAS USED 596993	VALUE 0	

Now we switch back to the Kovan Test Network in MetaMask, successfully compile the code, switch the Remix ENVIRONMENT to Injected Web3, enter the recipient address, and "transact" 0 wei to deploy the contract:



DEPLOY & RUN TRANSACTIONS



ENVIRONMENT



Injected Web3



Kovan (42) network



ACCOUNT



0xf8e...Aea4D (0.76724159)



GAS LIMIT



3000000

VALUE



0

wei



CONTRACT

DeferredEquityPlan - browser/Defer



DEPLOY



_EMPLOYEE: "0x2997608c322C755b06316!



Deploy



transact

Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 1

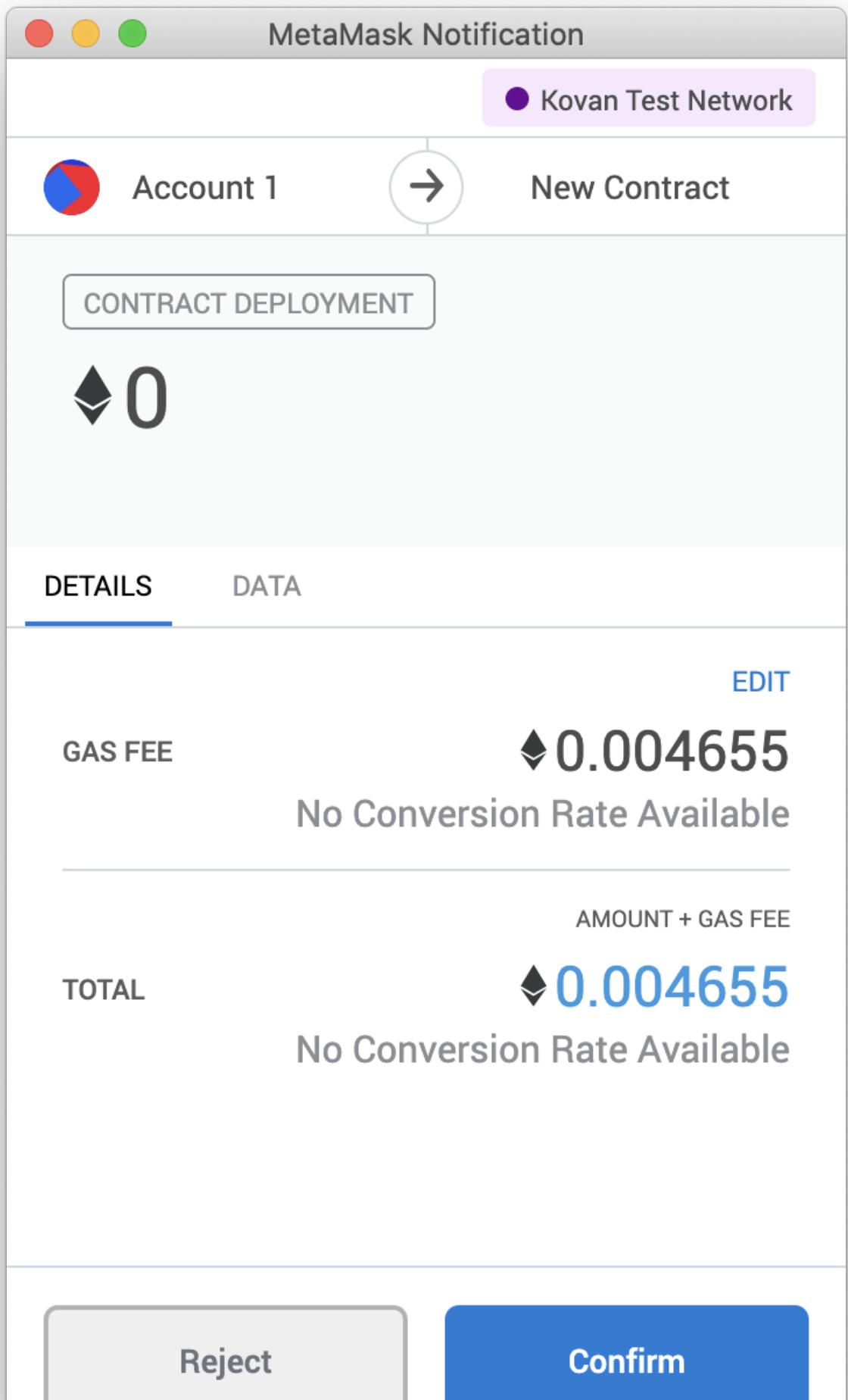


Deployed Contracts



Currently you have no contract instances
to interact with.

We Confirm in MetaMask:



The testnet address for others to be able to send to is:
0xf8eF4E21f1501EA4cCfE1110D5541545593Aea4D.