

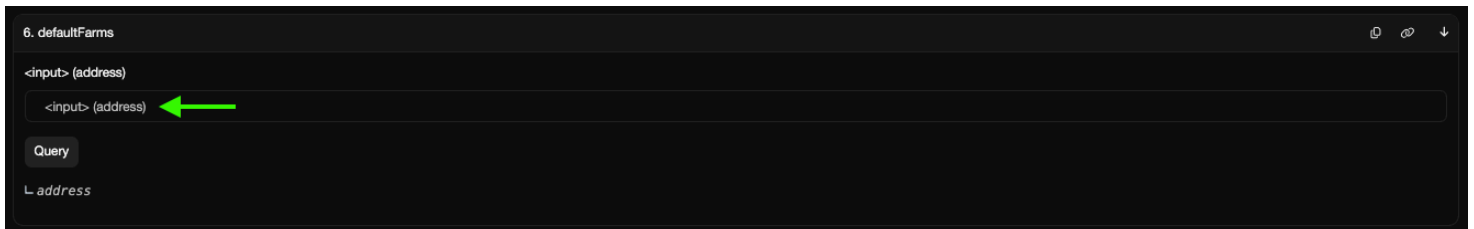
Initializing Farm Reward Distribution

Overview

As a resolver, offering farm incentives to delegators can help increase your Unicorn Power balance and consequently, your probability of filling orders. Upon registering as a resolver, the farm contract is automatically deployed. You can fetch the farm by interacting with the `dst1inch` contract's ABI. Note that you can reward delegators with any ERC20 token, not only tokens from the incentive program.

How to Fetch a Farm with the `dst1inch` Contract

1. Go to the "Read Contract" section of the `dst1inch` contract on Etherscan. Under the `defaultFarms` method, enter your resolver address within the 'address' parameter and send the call.



6. defaultFarms

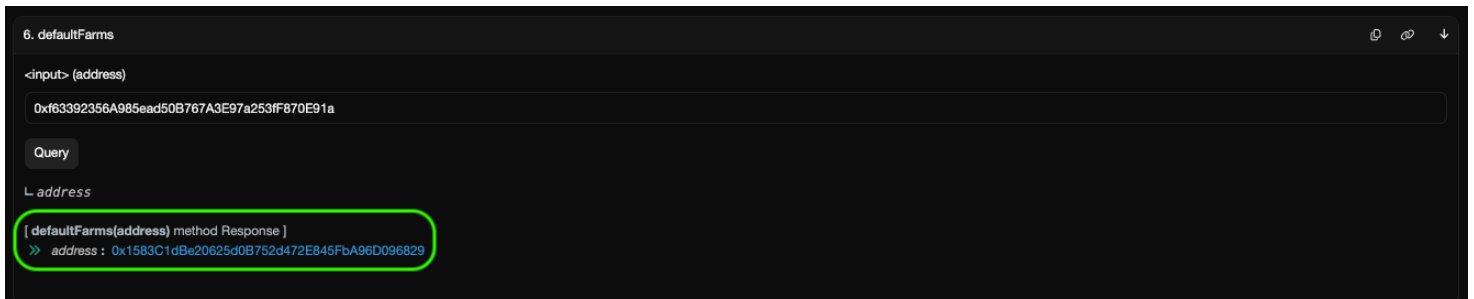
<input> (address)

<input> (address)

Query

L address

1. Click on the returned farm contract address and navigate to the 'Write Contract' section of the returned contract page. This is your newly generated farm distribution address.



6. defaultFarms

<input> (address)

0xf63392356A985ead50B767A3E97a253fF870E91a

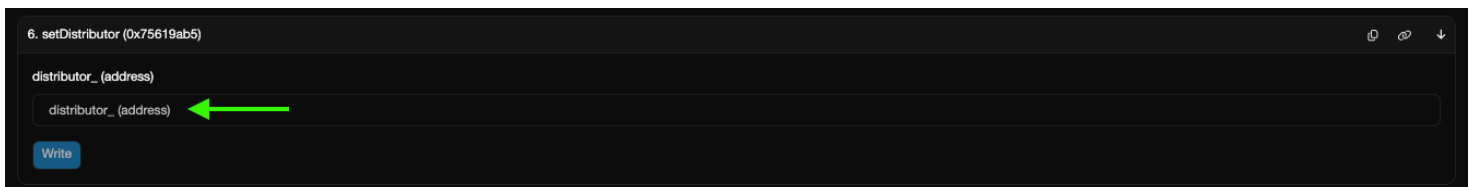
Query

L address

[defaultFarms(address) method Response]

>> address : 0x1583C1dBe20625d0B752d472E845FbA96D096829

3. Call the `setDistributor` method using the address that will be managing and distributing rewards.



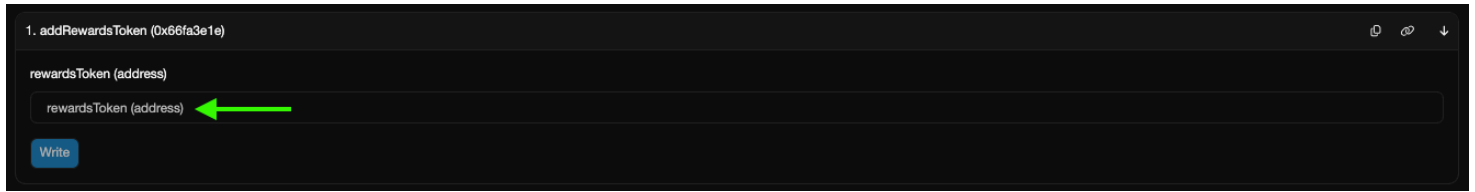
6. setDistributor (0x75619ab5)

distributor_ (address)

distributor_ (address)

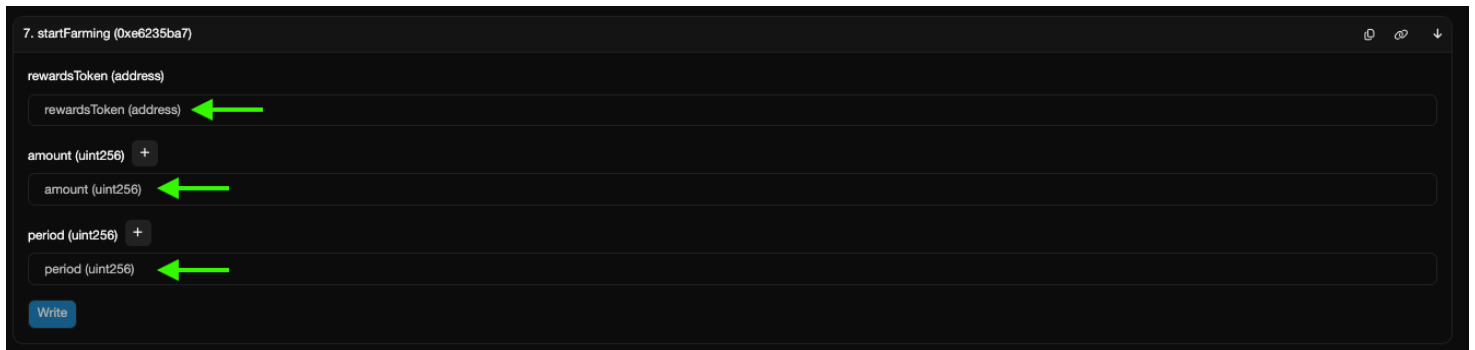
Write

4. On the same contract, call `addRewardsToken` with the desired token address that will be distributed to your delegators.



The screenshot shows the '1. addRewardsToken (0x66fa3e1e)' function interface. It has a single input field labeled 'rewardsToken (address)' with a green arrow pointing to it. Below the input field is a blue 'Write' button.

5. Lastly, on the same contract, call `startFarming`, entering the `rewardsToken` address, amount, and period (both amount and period are uint256). Once called, your farm will have started, and rewards will begin to be distributed to your delegators.



The screenshot shows the '7. startFarming (0xe6235ba7)' function interface. It has three input fields: 'rewardsToken (address)', 'amount (uint256)', and 'period (uint256)'. Each input field has a green arrow pointing to it. Below the input fields is a blue 'Write' button.

How to Replenish Farming Token Balance (Optional)

To replenish the token balance of your farm rewards, call `addRewardsToken` and `startFarming` to the farm contract as done in the initial setup.

Example ABI Interaction

! INFO

The example script below reads ABI definitions for both the `dst1inch` and `multiFarmingPod` contracts.

You will need to create two new files and add the ABI definitions which can be found at these URLs:

- `dst1inchABI.json`
- `abi.json`

(Please replace 'YourApiKeyToken' with your actual Etherscan API key at the end of each linked URL above)

```
require("dotenv").config(); //for accessing sensitive information such as private keys, API keys, etc.
```

```
const {Web3} = require("web3");  
const web3 = new Web3(`Your_ethereum_RPC`);  
const fs = require('fs');
```

```

const dst1inchABI = JSON.parse(fs.readFileSync('dst1inchABI.json', 'utf8'));
const farmABI = JSON.parse(fs.readFileSync('abi.json', 'utf8'));

const erc20Abi = [
  // ERC20 ABI fragment (for contract approval)
  {
    "constant": false,
    "inputs": [
      {
        "name": "spender",
        "type": "address"
      },
      {
        "name": "value",
        "type": "uint256"
      }
    ],
    "name": "approve",
    "outputs": [
      {
        "name": "",
        "type": "bool"
      }
    ],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "function"
  }
];

const dst1inch = "0xAccfAc2339e16DC80c50d2fa81b5c2B049B4f947";
const dst1inchContract = new web3.eth.Contract(dst1inchABI, dst1inch);

const account = 'YOUR_REGISTERED_ADDRESS';
const privateKey = 'YOUR_PRIVATE_KEY';

async function distributeFarmingRewards(resolverAddress, newDistributor, rewardsToken, amount, period) {
  try {
    // Call defaultFarms with your resolver address
    const farmAddress = await dst1inchContract.methods.defaultFarms(resolverAddress).call();
    console.log(`Farm address: ${farmAddress}`);

    const farmContract = new web3.eth.Contract(farmABI, farmAddress);
    const tokenContract = new web3.eth.Contract(erc20Abi, rewardsToken);

    // Call setDistributor on the returned farm address
    const setDistributorTx = {
      from: resolverAddress,
      to: farmAddress,
      data: farmContract.methods.setDistributor(newDistributor).encodeABI(),
    };

    const setDistributorSignedTx = await web3.eth.accounts.signTransaction(setDistributorTx, privateKey);
    await web3.eth.sendSignedTransaction(setDistributorSignedTx.rawTransaction);
    console.log("Distributor set successfully");

    // Approve the farm contract to spend the tokens
    const approveTx = {

```

```

    from: newDistributor,
    to: rewardsToken,
    data: tokenContract.methods.approve(farmAddress, amount).encodeABI(),
  };

  const approveSignedTx = await web3.eth.accounts.signTransaction(approveTx, privateKey);
  await web3.eth.sendSignedTransaction(approveSignedTx.rawTransaction);
  console.log("Approval transaction confirmed");

  // Call addRewardsToken with the address of distribution token
  const addRewardsTokenTx = {
    from: newDistributor,
    to: farmAddress,
    data: farmContract.methods.addRewardsToken(rewardsToken).encodeABI(),
  };

  const addRewardsTokenSignedTx = await web3.eth.accounts.signTransaction(addRewardsTokenTx, privateKey);
  await web3.eth.sendSignedTransaction(addRewardsTokenSignedTx.rawTransaction);
  console.log("Rewards token added successfully");

  // Call startFarming(address rewardsToken, uint256 amount, uint256 period)
  const startFarmingTx = {
    from: newDistributor,
    to: farmAddress,
    data: farmContract.methods.startFarming(rewardsToken, amount, period).encodeABI(),
  };

  const startFarmingSignedTx = await web3.eth.accounts.signTransaction(startFarmingTx, privateKey);
  await web3.eth.sendSignedTransaction(startFarmingSignedTx.rawTransaction);
  console.log("Farming started successfully");
} catch (error) {
  console.error("Error distributing farming rewards:", error);
}
}

const resolverAddress = account;
const newDistributor = account; // you can use any address as the distributor
const rewardsToken = '0xRewardsTokenAddress';
const amount = web3.utils.toWei("Amount", "Token Name"); // Amount of rewards tokens
const period = 3600; // Period in seconds

distributeFarmingRewards(resolverAddress, newDistributor, rewardsToken, amount, period)
  .then(() => console.log("Farming rewards distributed successfully"))
  .catch((error) => console.error("Error distributing farming rewards:", error));

```

Have questions? Please reach out to us in the live support chat!

Previous

< Fusion+ test examples

Next

Contract interaction >