

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.28+commit.7893614a.js

DEPLOY & RUN TRANSACTIONS

addEmployee

worker: 0x5838Da6a701c568545dCf80E
salary: 3500

Calldata Parameters **transact**

deactivateEm... address_worker
fundCompan...
payEmployees
terminateCon...
updatePayme... uint256_newInterval
companyAcc
companyBal
getEmployees

0: uint256: 500300
0: tuple(uint256,address,uint256,uint256,bool): 0,0x5838Da6a701c568545dCf80E,3500,1737387424,true

```
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87
```

```
function fundCompanyBalance() external payable {  
    require(msg.sender != companyAcc, "Company owner cannot fund directly");  
    companyBal += msg.value;  
    emit CompanyFunded(msg.sender, msg.value, companyBal);  
}  
  
function updatePaymentInterval(uint256 _newInterval) external onlyCompanyOwner {  
    require(_newInterval > 0, "Payment interval must be greater than 0");  
    paymentInterval = _newInterval;  
    emit PaymentIntervalUpdated(_newInterval);  
}  
  
function payEmployees() external onlyCompanyOwner {  
    require(block.timestamp >= lastPaymentTime + paymentInterval, "Payment interval has not elapsed");  
    require(companyBal >= totalSalary, "Insufficient company balance");  
  
    for (uint256 i = 0; i < employees.length; i++) {  
        if (employees[i].isActive) {  
            _sendPayment(employees[i].worker, employees[i].salary);  
            lastPaid[employees[i].worker] = block.timestamp;  
        }  
    }  
}
```

0 Listen on all transactions Filter with transaction hash or address

[vm] from: 0x583...eddC4 to: Payroll.(constructor) value: 0 wei data: 0x608...c0033 logs: 0
hash: 0x7b3...6ad02
transact to Payroll.addEmployee pending ...

[vm] from: 0x583...eddC4 to: Payroll.addEmployee(address,uint256) 0x5A8...C4d01 value: 0 wei
data: 0xe7f...00dac logs: 1 hash: 0xfdf...0440e
call to Payroll.getEmployees

[call] from: 0x5838Da6a701c568545dCf80E to: Payroll.getEmployees() data: 0x4ab...efa36

Scam Alert Initialize as git repo Did you know? You can use the help of AI for Solidity error, click on 'Ask RemixAI'. RemixAI Copilot (enabled)

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.28+commit.7893614a.js

DEPLOY & RUN TRANSACTIONS

addEmployee

worker: 0x5838Da6a701c568545dCf80E
salary: 3500

Calldata Parameters **transact**

deactivateEm... address_worker
fundCompan...
payEmployees
terminateCon...
updatePayme... uint256_newInterval
companyAcc
companyBal
getEmployees

0: uint256: 500300
0: tuple(uint256,address,uint256,uint256,bool): 0,0x5838Da6a701c568545dCf80E,3500,1737387424,true

```
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87
```

```
function fundCompanyBalance() external payable {  
    require(msg.sender != companyAcc, "Company owner cannot fund directly");  
    companyBal += msg.value;  
    emit CompanyFunded(msg.sender, msg.value, companyBal);  
}  
  
function updatePaymentInterval(uint256 _newInterval) external onlyCompanyOwner {  
    require(_newInterval > 0, "Payment interval must be greater than 0");  
    paymentInterval = _newInterval;  
    emit PaymentIntervalUpdated(_newInterval);  
}  
  
function payEmployees() external onlyCompanyOwner {  
    require(block.timestamp >= lastPaymentTime + paymentInterval, "Payment interval has not elapsed");  
    require(companyBal >= totalSalary, "Insufficient company balance");  
  
    for (uint256 i = 0; i < employees.length; i++) {  
        if (employees[i].isActive) {  
            _sendPayment(employees[i].worker, employees[i].salary);  
            lastPaid[employees[i].worker] = block.timestamp;  
        }  
    }  
}
```

0 Listen on all transactions Filter with transaction hash or address

transact to Payroll.fundCompanyBalance pending ...

[vm] from: 0xab8...35cb2 to: Payroll.fundCompanyBalance() 0x5A8...C4d01 value: 300 wei data: 0xca7...92e7c
logs: 1 hash: 0x978...30d1d
call to Payroll.companyBal

[call] from: 0xab8483f64d9c6d1EcF9b849Ae677d0315835cb2 to: Payroll.companyBal() data: 0xcd3...dbe46
transact to Payroll.payEmployees pending ...

Scam Alert Initialize as git repo Did you know? You can use the help of AI for Solidity error, click on 'Ask RemixAI'. RemixAI Copilot (enabled)

DEPLOY & RUN TRANSACTIONS

addEmployee

_worker: 0x58380a6a701c568545dcf6b0c

_salary: 3500

CalldataParameterstransact

deactivateEm...address_worker

fundCompan...

payEmployees

terminateCon...

updatePayme...uint256_newInterval

companyAcc

companyBal

getEmployees

0: tuple(uint256,address,uint256,uint256,bool) 0: 0x58380a6a701c568545dcf6b03fc6875f56beddc435001737387424tr ue

ThresholdSubMapper

Payroll.sol 1 x

```
64
65 function fundCompanyBalance() external payable { @infinite gas
66     require(msg.sender != companyAcc, "Company owner cannot fund directly");
67     companyBal += msg.value;
68     emit CompanyFunded(msg.sender, msg.value, companyBal);
69 }
70
71 function updatePaymentInterval(uint256_newInterval) external onlyCompanyOwner { @infinite gas
72     require(_newInterval > 0, "Payment interval must be greater than 0");
73     paymentInterval = _newInterval;
74     emit PaymentIntervalUpdated(_newInterval);
75 }
76
77 function payEmployees() external onlyCompanyOwner { @infinite gas
78     require(block.timestamp >= lastPaymentTime + paymentInterval, "Payment interval has not elapsed");
79     require(companyBal >= totalSalary, "Insufficient company balance");
80
81     for (uint256 i = 0; i < employees.length; i++) {
82         if (employees[i].isActive) {
83             _sendPayment(employees[i].worker, employees[i].salary);
84             lastPaid[employees[i].worker] = block.timestamp;
85         }
86     }
87 }
```

0 Listen on all transactions Filter with transaction hash or address

transact to Payroll.fundCompanyBalance pending ...

✓ [vm] from: 0xAb8...35cb2 to: Payroll.fundCompanyBalance() 0x5A8...C4d01 value: 500000 wei

data: 0xca7...92e7c logs: 1 hash: 0xb75...4e64b

call to Payroll.companyBal

Debug

CALL [call] from: 0xAb8483f64d9C6d1Ecf9b849Ae677D03315835cb2 to: Payroll.companyBal() data: 0xcd3...dbe46

transact to Payroll.payEmployees pending ...

Debug