

SponsorWhitelistControl

Conflux Dapps Sponsor

Message Call A C B f B C, Conflux A B A B B C B /

SponsorControl

- sponsor_for_gas
- sponsor_for_collateral
- sponsor_balance_for_gas
- sponsor_balance_for_collateral
- sponsor_limit_for_gas_fee
- whitelist

- *: sponsor_for_gas whitelist sponsor_limit_for_gas_fee sponsor_balance_for_gas sponsor_balance_for_gas
- : sponsor_balance_for_collateral whitelist sponsor_balance_for_collateral

sponsor_for_gas sponsor_for_collateral sponsor_limit_for_gas_fee SponsorCo

sponsor_for_gas setSponsorForGas(address contractAddr, uint upperBound)

1. sponsor_balance_for_gas
2. sponsor_limit_for_gas_fee upperBound sponsor_balance_for_gas
3. 1000 1000

sponsor_balance_for_gas sponsor_for_gas sponsor_balance_for_gas sponsor_for_gas
sponsor_limit_for_gas_fee

sponsor_for_collateral setSponsorForCollateral(address contractAddr) sponsor_for_collateral
sponsor_balance_for_collateral

```
setSponsorForGas(address contractAddr, uint upperBound) setSponsorForCollateral( address
contractAddr) sponsor_limit_for_gas_fee
```

```
addPrivilege(address[] memory) sponsor_for_gas 0x0000000000000000000000000000000000000000000000000000000000000000
removePrivilege(address[] memory)
```

1.

```
addPrivilegeByAdmin(address contractAddr, address[] memory addresses)
removePrivilegeByAdmin(address contractAddr, address[] memory addresses)
```

```
pragma solidity >=0.4.15;

import "https://github.com/Conflux-Chain/conflux-
rust/blob/master/internal_contract/contracts/SponsorWhitelistControl.sol";

contract CommissionPrivilegeTest {
    mapping(uint => uint) public ss;

    function add(address account) public payable {
        SponsorWhitelistControl cpc =
SponsorWhitelistControl(0x0888000000000000000000000000000000000000000000000000000000000001);
        address[] memory a = new address[](1);
```

```

        a[0] = account;
        cpc.addPrivilege(a);
    }

    function remove(address account) public payable {
        SponsorWhitelistControl cpc =
SponsorWhitelistControl(0x0888000000000000000000000000000000000001);
        address[] memory a = new address[](1);
        a[0] = account;
        cpc.removePrivilege(a);
    }

    function foo() public payable {
    }

    function par_add(uint start, uint end) public payable {
        for (uint i = start; i < end; i++) {
            ss[i] = 1;
        }
    }
}

```

contract_addr

/

```

const PRIVATE_KEY = '0xxxxxxx';
const cfx = new Conflux({
    url: 'http://test.confluxrpc.org',
    logger: console,
});

const account = cfx.wallet.addPrivateKey(PRIVATE_KEY); // create account instance

const sponsor_contract = cfx.InternalContract('SponsorWhitelistControl');
sponsor_contract.setSponsorForGas(contract_addr, your_upper_bound).sendTransaction({
    from: account,
    value: your_sponsor_value
}).confirmed();

```

setSponsorForGas(contract_addr, your_upper_bound)

setSponsorForCollateral(contract_addr)

addPrivilege

removePrivilege

whitelist

0x00

whitelist

```
you_contract.add(white_list_addr).sendTransaction({
  from: account,
})

you_contract.remove(white_list_addr).sendTransaction({
  from: account,
})
```

`whitelist` `you_contract.foo()` `you_contract.par_add(1, 10)`

Revision #2

Created 14 June 2022 00:37:51 by Pana

Updated 18 July 2022 02:11:43 by Darwin