

基础实验

练习1

1. 应该在管理者函数即发起新会议newConference以及销毁合约destruct指定管理员身份。

```
modifier onlyOwner() {  
    require(msg.sender == administrator);  
    _;  
}
```

2. 通过require验证函数调用者地址是否是管理员的地址来确定。

require、assert、revert的区别：require、revert在不满足条件时会退还剩下的gas而assert会烧掉全部gas。revert、revert除了会把剩余gas退还，还允许返回一个值，assert不能。对于复杂的if/else逻辑流，一般考虑使用revert而不是require

3. Storage变量是指永久存储在区块链中的变量，而Memory变量则是临时的，当外部函数对某合约调用完成时，内存型变量即被移除

enrollment.sol代码

```
pragma solidity >=0.4.22 <0.9.0;  
pragma experimental ABIEncoderV2;  
  
contract Enrollment {  
    address public administrator;  
    struct Conference {  
        string title;  
        string detail;  
        uint limitation;  
        uint current;  
        bool isFull;  
    }  
    struct Participant {  
        string username;  
        string[] myconferences;  
    }  
    // 参与者address 对应 参与者  
    mapping(address => Participant) participants;  
    // 委托人address 对应 参与者  
    mapping(address => Participant[]) trustees;  
    Conference[] public conferences;  
  
    constructor() public {  
        administrator = msg.sender;  
    }  
}
```

```

modifier onlyOwner() {
    require(msg.sender == administrator);
    _;
}

event NewConference(string,string);
event ConferenceExpire(string);
event MyNewConference(string);

function signUp(string _username) public returns(string){
    require(bytes(participants[msg.sender].username).length<=0,"User
exists.");
    participants[msg.sender].username = _username;
    return participants[msg.sender].username;
}

function delegate(address trustee) public {
    require(trustee != msg.sender);
    trustees[trustee].push(participants[msg.sender]);
}

function enroll(string _title) public returns(string){
    bool finded = false;
    for(uint i = 0; i < conferences.length; ++i){
        if(keccak256(abi.encodePacked(_title)) ==
keccak256(abi.encodePacked(conferences[i].title))){
            if(!conferences[i].isFull){
                participants[msg.sender].myconferences.push(_title);
                conferences[i].current++;
                if(conferences[i].current == conferences[i].limitation){
                    conferences[i].isFull = true;
                }
                emit MyNewConference(_title);
                finded = true;
                break;
            }
            emit ConferenceExpire(_title);
        }
    }
    if(!finded){
        revert('fail to enroll');
    }
    return _title;
}

function enrollFor(string _title, string _username) public returns(string){
    bool finded = false;
    uint userId = 0;
    for(uint i = 0; i < trustees[msg.sender].length; ++i){
        if(keccak256(abi.encodePacked(trustees[msg.sender]
[i].username)) == keccak256(abi.encodePacked(_username))){
            finded = true;
            userId = i;
        }
    }
    if(!finded){
        revert('fail to enroll');
    }
}

```

```

    }
    finded = false;
    for(i = 0; i < conferences.length; ++i){
        if(keccak256(abi.encodePacked(_title)) ==
keccak256(abi.encodePacked(conferences[i].title))){
            if(!conferences[i].isFull){
                trustees[msg.sender][userId].myconferences.push(_title);
                conferences[i].current++;
                if(conferences[i].current == conferences[i].limitation){
                    conferences[i].isFull = true;
                }
                finded = true;
                break;
            }
        }
    }
    if(!finded){
        revert('fail to enroll');
    }
    return _title;
}

function newConference(string _title, string _detail,uint _limitation)
public onlyOwner returns(string){
    conferences.push(Conference(_title, _detail,_limitation, 0,false));
    emit NewConference(_title,_detail);
    return _title;
}

function queryConfList() public view returns(string[], string[]) {
    string[] memory title = new string[](conferences.length);
    string[] memory detail = new string[](conferences.length);
    uint count = 0;
    for(uint i = 0; i < conferences.length; ++i){
        title[count] = conferences[i].title;
        detail[count] = conferences[i].detail;
        count++;
    }
    return (title, detail);
}

function queryMyConf() public view returns(string[]) {
    return participants[msg.sender].myconferences;
}

function destruct() private onlyOwner{
    selfdestruct(administrator);
}

}

```

练习2

测试文件进行测试

```
$ truffle test
Using network 'development'.

Compiling your contracts...
=====
> Compiling ./test/test.sol
> Compilation warnings encountered:

    project:/contracts/Enrollment.sol:2:1: Warning: Experimental
pragma experimental ABIEncoderV2;
^-----^
,project:/test/test.sol:8:5: Warning: No visibility specified
    function testAdministratorIdentification() {
      ^ (Relevant source part starts here and spans across multiple
lines).

> Artifacts written to /tmp/test--4544-DfFxysVBnPkP
> Compiled successfully using:
   - solc: 0.4.26+commit.4563c3fc.Emscripten.clang

TestEnrollment
  ✓ testAdministratorIdentification (193ms)
  ✓ testSignUp (238ms)
  ✓ testNewConf (211ms)
  ✓ testEnroll (246ms)

4 passing (8s)
```

合约部署

```

L$ truffle migrate

Compiling your contracts...
=====
> Everything is up to date, there is nothing to compile.

Starting migrations...
=====
> Network name: 'development'
> Network id: 5777
> Block gas limit: 6721975 (0x6691b7)

1_initial_migration.js
=====

Replacing 'Migrations'
-----
> transaction hash: 0x61db8df7ec25ebebdf4a0f6e206d410666263e74fa57b46b8021a65f63e40a02
> Blocks: 0 Seconds: 0
> contract address: 0x3906f3d59785A2305a12eF2Ca181eC3Dd26F2463
> block number: 1
> block timestamp: 1637664895
> account: 0xB18377d9d09c35f341BA2C021EBc3B1ECD3Ce819
> balance: 99.99592882
> gas used: 203559 (0x31b27)
> gas price: 20 gwei
> value sent: 0 ETH
> total cost: 0.00407118 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost: 0.00407118 ETH

2_deploy_contracts.js
=====

Replacing 'ConvertLib'
-----
> transaction hash: 0x67d8ca500c052b0b2d5546b160f42ab19758faa0ccdde20a21f9f025d88ce239
> Blocks: 0 Seconds: 0
> contract address: 0x2Ac4E72e4b1c1AbdB52Be437f126531Ddd161A0B
> block number: 3
> block timestamp: 1637664896
> account: 0xB18377d9d09c35f341BA2C021EBc3B1ECD3Ce819
> balance: 99.99306858
> gas used: 100683 (0x1894b)
> gas price: 20 gwei
> value sent: 0 ETH
> total cost: 0.00201366 ETH

Replacing 'Enrollment'
-----
> transaction hash: 0xb9bbf4b999de23c602a88c718d38628a5cc756155eb96f46efd6da7226a4e7dd
> Blocks: 0 Seconds: 0
> contract address: 0xEE6D2d793939Aa6730cd8C4e9DdEd583ae13144d
> block number: 4
> block timestamp: 1637664896
> account: 0xB18377d9d09c35f341BA2C021EBc3B1ECD3Ce819
> balance: 99.95658924
> gas used: 1823967 (0x1bd4df)
> gas price: 20 gwei
> value sent: 0 ETH
> total cost: 0.03647934 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost: 0.038493 ETH

Summary
=====
> Total deployments: 3
> Final cost: 0.04256418 ETH

```

部署后Ganache相关记录

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK
5

GAS PRICE
2000000000

GAS LIMIT
6721975

HARDFORK
MUIRGLACIER

NETWORK ID
5777

RPC SERVER
HTTP://127.0.0.1:7545

MINING STATUS
AUTOMINING

WORKSPACE
PROUD-THUNDER

SWITCH

BLOCK
5

MINED ON
2021-11-23 18:54:56

GAS USED
27329

1 TRANSACTION

BLOCK
4

MINED ON
2021-11-23 18:54:56

GAS USED
1823967

1 TRANSACTION

BLOCK
3

MINED ON
2021-11-23 18:54:56

GAS USED
100683

1 TRANSACTION

BLOCK
2

MINED ON
2021-11-23 18:54:56

GAS USED
42329

1 TRANSACTION

BLOCK
1

MINED ON
2021-11-23 18:54:55

GAS USED
203559

1 TRANSACTION

BLOCK
0

MINED ON
2021-11-23 18:54:47

GAS USED
0

NO TRANSACTIONS

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK5

GAS PRICE2000000000

GAS LIMIT6721975

HARDFORKMUIRGLACIER

NETWORK ID5777

RPC SERVERHTTP://127.0.0.1:7545

MINING STATUSAUTOMINING

WORKSPACEPROUD-THUNDER

SWITCH

TX HASH

0x783a2ccbda6526c26d834dc8db959ef11c40a1188722a667b26ac83584467c4e

CONTRACT CALL

FROM ADDRESS

0xB18377d9d09c35f341BA2C021EBc3B1ECD3Ce819

TO CONTRACT ADDRESS

Migrations

GAS USED

27329

VALUE

0

TX HASH

0xb9bbf4b999de23c602a88c718d38628a5cc756155eb96f46efd6da7226a4e7dd

CONTRACT CREATION

FROM ADDRESS

0xB18377d9d09c35f341BA2C021EBc3B1ECD3Ce819

CREATED CONTRACT ADDRESS

0xEE6D2d793939Aa6730cd8C4e9DdEd583ae13144d

GAS USED

1823967

VALUE

0

TX HASH

0x67d8ca500c052b0b2d5546b160f42ab19758faa0ccdde20a21f9f025d88ce239

CONTRACT CREATION

FROM ADDRESS

0xB18377d9d09c35f341BA2C021EBc3B1ECD3Ce819

CREATED CONTRACT ADDRESS

0x2Ac4E72e4b1c1AbdB52Be437f126531Ddd161A0B

GAS USED

100683

VALUE

0

TX HASH

0xf4507a02c215def46d6b032c28991b7f0a9f1c07c043bb5fd13fb2e3f623d413

CONTRACT CALL

FROM ADDRESS

0xB18377d9d09c35f341BA2C021EBc3B1ECD3Ce819

TO CONTRACT ADDRESS

Migrations

GAS USED

42329

VALUE

0

TX HASH

0x61db8df7ec25ebdbdf4a0f6e206d410666263e74fa57b46b8021a65f63e40a02

CONTRACT CREATION

FROM ADDRESS

0xB18377d9d09c35f341BA2C021EBc3B1ECD3Ce819

CREATED CONTRACT ADDRESS

0x3906f3d59785A2305a12eF2Ca181eC3Dd26F2463

GAS USED

203559

VALUE

0

lab8 /home/samrito/Documents/GO/BlockChain/Lab6/lab8

NAME ConvertLib	ADDRESS 0x2Ac4E72e4b1c1AbdB52Be437f126531Ddd161A0B	TX COUNT 0	DEPLOYED
NAME Enrollment	ADDRESS 0xEE6D2d793939Aa6730cd8C4e9DdEd583ae13144d	TX COUNT 0	DEPLOYED
NAME Migrations	ADDRESS 0x3906f3d59785A2305a12eF2Ca181eC3Dd26F2463	TX COUNT 1	DEPLOYED

合约部署流程以及合约调用

首先编译合约，依据1_initial_migration.js初始化部署，调用Migrations合约。通过区块1可以看到包含CONTRACT CREATION账单，之后则是区块2 CONTRACT CALL账单，调用Migrations合约的setCompleted函数，通知初始化部署完成

之后是2_deploy_contracts.js部署合约ConvertLib和Enrollment合约，再次调用Migrations合约通知部署完成。区块3,4包含CONTRACT CREATION账单分别部署ConvertLib以及Enrollment合约。之后则是区块5包含CONTRACT CALL账单，调用Migrations合约的setCompleted函数

测试文件代码：

```

pragma solidity >=0.4.22 <0.9.0 ;

import "truffle/Assert.sol";
import "truffle/DeployedAddresses.sol";
import "../contracts/Enrollment.sol";

contract TestEnrollment{
    function testAdministratorIdentification() {
        Enrollment enroll = Enrollment(DeployedAddresses.Enrollment());
        Assert.equal(address(enroll.administrator),
DeployedAddresses.Enrollment(), 'administior error');
    }
    function testSignUp() public {
        Enrollment en = new Enrollment();
        string memory name = en.signUp("alice");
        string memory expected_name = "alice";
        Assert.equal(name,expected_name,"signup failed");
    }
    function testNewConf() public {
        Enrollment enroll = new Enrollment();
        string memory expected = "conf1";
        Assert.equal(enroll.newConference("conf1","beijing",30),expected,"new
conference failed");
    }
    function testEnroll() public {
        Enrollment enroll = new Enrollment();
        enroll.newConference("conf1", "beijing", 30);
        string memory expected = "conf1";
        Assert.equal(enroll.enroll("conf1"),expected,"enroll failed");
    }
}

```

练习3

首先完成abi调用信息补全以及合约地址填写

```

import Web3 from 'web3';
//在此粘贴ABI信息
const abi = [
  {
    "constant": true,
    "inputs": [
      {
        "name": "",
        "type": "uint256"
      }
    ],
    "name": "conferences",
    "outputs": [
      {
        "name": "title",
        "type": "string"
      },
      {

```

```

        "name": "detail",
        "type": "string"
    },
    {
        "name": "limitation",
        "type": "uint256"
    },
    {
        "name": "current",
        "type": "uint256"
    },
    {
        "name": "isFull",
        "type": "bool"
    }
],
"payable": false,
"stateMutability": "view",
"type": "function"
},
{
    "constant": true,
    "inputs": [],
    "name": "administrator",
    "outputs": [
        {
            "name": "",
            "type": "address"
        }
    ],
    "payable": false,
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "constructor"
},
{
    "anonymous": false,
    "inputs": [
        {
            "indexed": false,
            "name": "",
            "type": "string"
        },
        {
            "indexed": false,
            "name": "",
            "type": "string"
        }
    ],
    "name": "NewConference",
    "type": "event"
},
{

```



```

    "anonymous": false,
    "inputs": [
      {
        "indexed": false,
        "name": "",
        "type": "string"
      }
    ],
    "name": "ConferenceExpire",
    "type": "event"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": false,
        "name": "",
        "type": "string"
      }
    ],
    "name": "MyNewConference",
    "type": "event"
  },
  {
    "constant": false,
    "inputs": [
      {
        "name": "_username",
        "type": "string"
      }
    ],
    "name": "signUp",
    "outputs": [
      {
        "name": "",
        "type": "string"
      }
    ],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "constant": false,
    "inputs": [
      {
        "name": "trustee",
        "type": "address"
      }
    ],
    "name": "delegate",
    "outputs": [],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "constant": false,

```

```

"inputs": [
  {
    "name": "_title",
    "type": "string"
  }
],
"name": "enroll",
"outputs": [
  {
    "name": "",
    "type": "string"
  }
],
"payable": false,
"stateMutability": "nonpayable",
"type": "function"
},
{
  "constant": false,
  "inputs": [
    {
      "name": "_title",
      "type": "string"
    },
    {
      "name": "_username",
      "type": "string"
    }
  ],
  "name": "enrollFor",
  "outputs": [
    {
      "name": "",
      "type": "string"
    }
  ],
  "payable": false,
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "constant": false,
  "inputs": [
    {
      "name": "_title",
      "type": "string"
    },
    {
      "name": "_detail",
      "type": "string"
    },
    {
      "name": "_limitation",
      "type": "uint256"
    }
  ],
  "name": "newConference",
  "outputs": [

```

```

        {
            "name": "",
            "type": "string"
        }
    ],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "function"
},
{
    "constant": true,
    "inputs": [],
    "name": "queryConfList",
    "outputs": [
        {
            "name": "",
            "type": "string[]"
        },
        {
            "name": "",
            "type": "string[]"
        }
    ],
    "payable": false,
    "stateMutability": "view",
    "type": "function"
},
{
    "constant": true,
    "inputs": [],
    "name": "queryMyConf",
    "outputs": [
        {
            "name": "",
            "type": "string[]"
        }
    ],
    "payable": false,
    "stateMutability": "view",
    "type": "function"
}
];

// 在此粘贴合约地址
const address = '0x2727fbBD888059D391977305B884e3C5D0bF7Cf6';

// window.web3.currentProvider为当前浏览器的web3 Provider
const web3 = new Web3(window.ethereum);

// 允许连接到metamask
window.ethereum.enable();

// 导出合约实例
export default new web3.eth.Contract(abi, address);

```

组件代码完成

- signup

```
import React, { Component } from 'react'
import { Button, Checkbox, Form } from 'semantic-ui-react'
import { connect } from 'react-redux'
import contract from '../contracts/contract'

class SignUp extends (Component) {
  render() {
    return (
      <Form>
        <Form.Field>
          <input placeholder='username' value={this.props.username} onChange=
{this.props.handleChange} />
        </Form.Field>
        <Form.Field>
          <input placeholder='extra' value={this.props.extra} onChange=
{this.props.handleChange} />
        </Form.Field>
        <Button type='submit' onClick={() =>
{this.props.submit(this.props.username)}}>Submit</Button>
      </Form>
    )
  }
}

const mapStateToProps = (state) => {
  return (
    {
      username: state.signup.username,
      extra: state.signup.extra
    }
  )
}

const mapDispatchToProps = (dispatch) => {
  return {
    submit(username) { //请在line 16的onClick内传入参数
      contract.methods.signUp(username) //输入参数
      .send({from:window.web3.eth.accounts[0]},function(err,res)
{console.log(res)}) //function中的res为方法返回值
      .then((res)=>console.log(res)); //该res为交易执行完后的具体交易信息，如TxHash等
      dispatch({
        type: 'submit_signup'
      })
    },

    handleChange(e) {
      if (e.target.placeholder === 'username')
        dispatch({
          type: 'username',
          value: e.target.value
        })
      else
        dispatch({
          type: 'extra',
          value: e.target.value
        })
    }
  }
}
```

```

    })
  },
}

}

export default connect(mapStateToProps, mapDispatchToProps)(SignUp)

```

- delegate

```

/*
 * @Author: Samrito
 * @Date: 2021-11-19 19:39:36
 * @LastEditors: Samrito
 * @LastEditTime: 2021-11-22 23:01:53
 */
import React, { Component } from 'react'
import { Button, Form } from 'semantic-ui-react'
import { connect } from 'react-redux'
import contract from '../../contracts/contract'

class Delegate extends (Component) {
  render() {
    return (
      <Form>
        <Form.Field>
          <input placeholder="trustee's address" value={this.props.address}
onChange={this.props.handleChange} />
        </Form.Field>
        <Button type='submit' onClick={()=>
{this.props.submit(this.props.address)}}>Submit</Button>
      </Form>
    )
  }
}

const mapStateToProps = (state) => {
  return (
    {
      address: state.delegate.address,
    }
  )
}

const mapDispatchToProps = (dispatch) => {
  return {
    submit(address) {
      contract.methods.delegate(address) //输入参数
      .send({from:window.web3.eth.accounts[0]},function(err,res)
{console.log(res)}) //function中的res为方法返回值
      .then((res)=>console.log(res));
      dispatch({
        type: 'submit_delegate'
      })
    },

    handleChange(e) {

```

```

        dispatch({
          type: 'address',
          value: e.target.value
        })
      },
    }
  }

export default connect(mapStateToProps, mapDispatchToProps)(Delegate)

```

- enroll

```

/*
 * @Author: Samrito
 * @Date: 2021-11-19 19:39:36
 * @LastEditors: Samrito
 * @LastEditTime: 2021-11-22 21:31:32
 */
import React, { Component } from 'react'
import { Button, Form } from 'semantic-ui-react'
import { connect } from 'react-redux'
import contract from '../contracts/contract'

class Enroll extends (Component) {
  render() {
    return (
      <Form>
        <Form.Field>
          <input placeholder="Title of Conference" value={this.props.title}
onChange={this.props.handleChange} />
        </Form.Field>
        <Button type='submit' onClick={() =>
{this.props.submit(this.props.title)}}>Submit</Button>
      </Form>
    )
  }
}

const mapStateToProps = (state) => {
  return (
    {
      title: state.enroll.title,
    }
  )
}

const mapDispatchToProps = (dispatch) => {
  return {
    submit(title) {
      contract.methods.enroll(title) //输入参数
      .send({from:window.web3.eth.accounts[0]},function(err,res)
{console.log(res)}) //function中的res为方法返回值
      .then((res)=>console.log(res));
      dispatch({
        type: 'submit_enroll'
      })
    }
  }
}

```

```

    },

    handleChange(e) {
      dispatch({
        type: 'enroll_title',
        value: e.target.value
      })
    },
  },
}

export default connect(mapStateToProps, mapDispatchToProps)(Enroll)

```

- enrollfor

```

/*
 * @Author: Samrito
 * @Date: 2021-11-19 19:39:36
 * @LastEditors: Samrito
 * @LastEditTime: 2021-11-22 21:41:08
 */
import React, { Component } from 'react'
import { Button, Form } from 'semantic-ui-react'
import { connect } from 'react-redux'
import contract from '../contracts/contract'

class EnrollFor extends (Component) {
  render() {
    return (
      <Form>
        <Form.Field>
          <input placeholder="Title of Conference" value={this.props.title}
            onChange={this.props.handleChange} />
        </Form.Field>
        <Form.Field>
          <input placeholder="Username" value={this.props.username} onChange=
            {this.props.handleChange} />
        </Form.Field>
        <Button type='submit' onClick={() =>
          {this.props.submit(this.props.title, this.props.username)}}>Submit</Button>
      </Form>
    )
  }
}

const mapStateToProps = (state) => {
  return (
    {
      title: state.enrollfor.title,
      username: state.enrollfor.username
    }
  )
}

const mapDispatchToProps = (dispatch) => {
  return {

```

```

    submit(title, username) {
      contract.methods.enrollFor(title, username) //输入参数
        .send({from: window.web3.eth.accounts[0]}, function(err, res)
{console.log(res)}) //function中的res为方法返回值
        .then((res) => console.log(res));
      dispatch({
        type: 'submit_enrollfor'
      })
    },

    handleChange(e) {
      if (e.target.placeholder === 'Title of Conference')
        dispatch({
          type: 'enrollfor_title',
          value: e.target.value
        })
      else
        dispatch({
          type: 'enrollfor_username',
          value: e.target.value
        })
    },
  },
}

export default connect(mapStateToProps, mapDispatchToProps)(EnrollFor)

```

- newconf

```

/*
 * @Author: Samrito
 * @Date: 2021-11-19 19:39:36
 * @LastEditors: Samrito
 * @LastEditTime: 2021-11-22 21:42:19
 */
import React, { Component } from 'react'
import { Button, Form } from 'semantic-ui-react'
import { connect } from 'react-redux'
import contract from "../../contracts/contract"

class NewConf extends Component {
  render() {
    return (
      <Form>
        <Form.Field>
          <input placeholder="Title" value={this.props.title} onChange=
{this.props.handleChange} />
        </Form.Field>
        <Form.Field>
          <input placeholder="Detail" value={this.props.detail} onChange=
{this.props.handleChange} />
        </Form.Field>
        <Form.Field>
          <input placeholder="Limitation" value={this.props.limitation}
onChange={this.props.handleChange} />

```



```

        </Form.Field>
        <Button type='submit' onClick={ () =>
{this.props.submit(this.props.title,this.props.detail,
this.props.limitation)}}>Submit</Button>
        </Form>
    )
  }
}

const mapStateToProps = (state) => {
  return (
    {
      title: state.newconf.title,
      detail: state.newconf.detail,
      limitation: state.newconf.limitation,
    }
  )
}

const mapDispatchToProps = (dispatch) => {
  return {
    submit(title, detail, limitation) {
      contract.methods.newConference(title,detail,limitation) //输入参数
      .send({from:window.web3.eth.accounts[0]},function(err,res)
{console.log(res)}) //function中的res为方法返回值
      .then((res)=>console.log(res));
      dispatch({
        type: 'submit_newconf'
      })
    },



    handleChange(e) {
      switch (e.target.placeholder) {
        case 'Title':
          dispatch({
            type: 'newconf_title',
            value: e.target.value
          })
          break;
        case 'Detail':
          dispatch({
            type: 'detail',
            value: e.target.value
          })
          break;
        case 'Limitation':
          dispatch({
            type: 'limitation',
            value: e.target.value
          })
          break;
        default:
          break;
      }
    },
  },
}
}

```

```
export default connect(mapStateToProps, mapDispatchToProps)(NewConf)
```

实验结果展示

- 建立两个账户，其中一个为administor，即如下两个地址


ADDRESS	BALANCE	TX COUNT	INDEX	
0xdC937646B829AD63578D417Ff979eb32348339D3	99.96 ETH	5	0	
0x04a5570B7969d8432E594F0C0df629FC856e92ae	100.00 ETH	0	1	


使用signup注册


Sign Up

Submit

My Conferences


 Ant Design Title 1

 Ant Design Title 2


 Ant Design Title 3

< 1 / 2 >


Conference List

 Ant Design Title 1


other information

 Ant Design Title 2


other information

 Ant Design Title 3

other information

 Ant Design Title 4

other information

 Ant Design Title 5

other information

< 1 / 2 >

New Conference

Submit

Enroll

Submit

Enroll For

Submit

Delegate

Submit

Delegate

0xdc937646B829AD63578D417Ff979eb3234

Submit

← BACK

TX 0xcacffdaa9e6024e2389541a9f386531495b211f29437f79c6383b2202696ac71

SENDER ADDRESS

0x04a5570B7969d8432E594F0C0df629FC856e92ae

TO CONTRACT ADDRESS

0x6D62831C9e4416250F24ab22F9C4a9b532D5c2b5

CONTRACT CALL

VALUE

0.00 ETH

GAS USED

68380

GAS PRICE

20000000000

GAS LIMIT

104512

MINED IN BLOCK

7

TX DATA

0x5c19a95c000000000000000000000000dc937646b829ad63578d417ff979eb32348339d3

CONTRACT

CONTRACT

Enrollment

ADDRESS

0x6D62831C9e4416250F24ab22F9C4a9b532D5c2b5

FUNCTION

delegate(trustee: address)

INPUTS

0xdc937646b829ad63578d417ff979eb32348339d3

EVENTS

- account2无法建立会议

?

Ganache

Account 5

→

0x6D62...c2b5

CONTRACT INTERACTION

0

DETAILS

DATA

GAS FEE

527.643222

No Conversion Rate Available

Gas Price (GWEI)

20

Gas Limit

6385876

AMOUNT + GAS FEE

TOTAL

527.643222

No Conversion Rate Available

!

ALERT: Transaction Error. Exception thrown in contract code.

Reject

Confirm

- account1建立会议

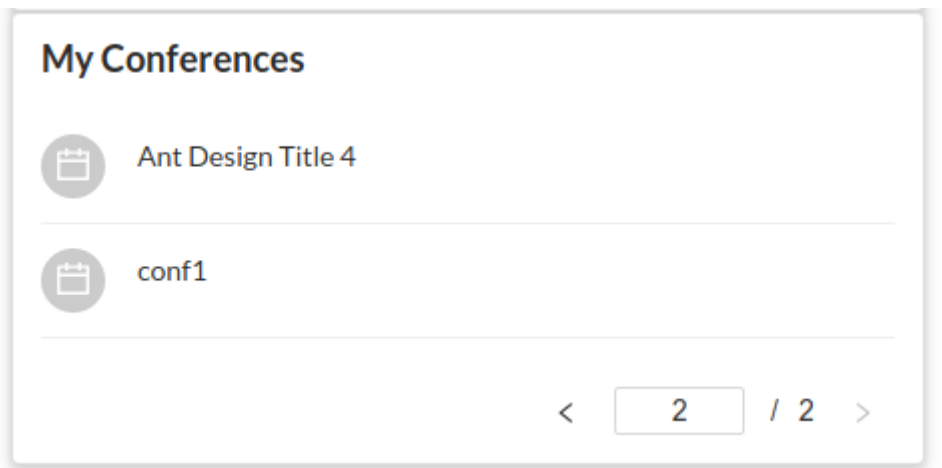
New Conference

conf1

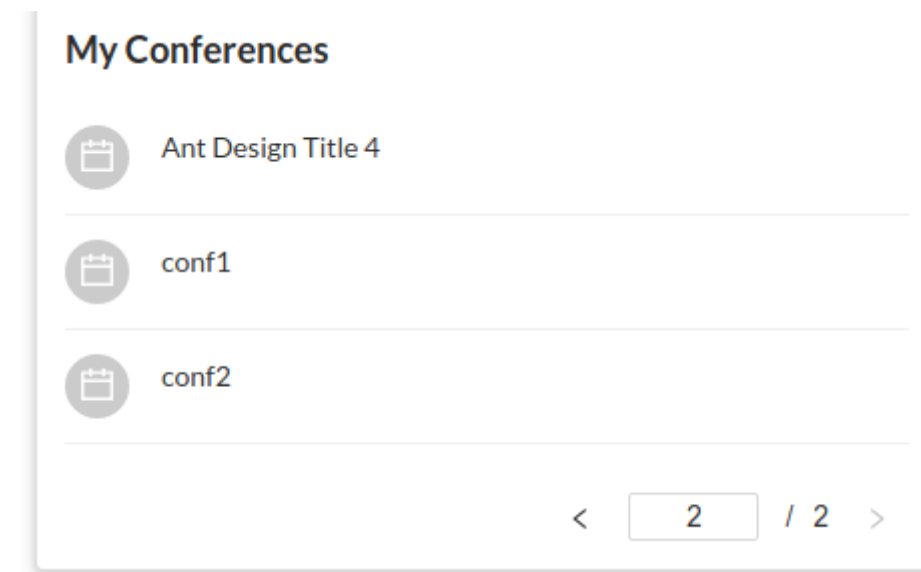
hi1

10

Submit



- account1报名conf2,在myconferences组建可以看到



实验代码

- conflist

```
/*
 * @Author: Samrito
 * @Date: 2021-11-19 19:39:36
 * @LastEditors: Samrito
 * @LastEditTime: 2021-11-23 16:26:48
 */
import React, { Component } from 'react'
import { List, Avatar } from 'antd'
import { CalendarOutlined } from '@ant-design/icons';
import 'antd/dist/antd.css'
import contract from '../contracts/contract'

const data = [
  {
    title: 'Ant Design Title 1',
    detail: 'other information'
  },
  {
    title: 'Ant Design Title 2',
    detail: 'other information'
  },
  {
```



```

        title: 'Ant Design Title 3',
        detail: 'other information'
    },
    {
        title: 'Ant Design Title 4',
        detail: 'other information'
    },
    {
        title: 'Ant Design Title 5',
        detail: 'other information'
    },
    {
        title: 'Ant Design Title 6',
        detail: 'other information'
    },
];

class ConfList extends (Component) {
  componentDidMount() {
    //先执行一遍查询操作
    contract.methods.queryConfList()
      .call({from:window.web3.eth.accounts[0]},(err,res)=>{
        this.setState({loading:true});
        for(var i = 0; i < res.length;i+=2){
          data.push({title:res[i],detail:res[i+1]});
        }
      }) //function中的res为方法返回值
      .then(()=>{
        this.setState({loading:false});
      });

    contract.events.NewConference({
      filter: {}, // Using an array means OR: e.g. 20 or 23
      fromBlock: window.web3.eth.getBlockNumber()
    }, (error, event)=>{
      this.setState({loading:true});
      data.push({title:event.returnValues[0],detail:event.returnValues[1]});
    }) //重新执行查询操作
  }

  render() {
    return (
      <List
        itemLayout="horizontal"
        dataSource={data}
        pagination={{ pageSize: 5, simple: 'true' }}
        renderItem={item => (
          <List.Item>
            <List.Item.Meta
              avatar={<Avatar icon={<CalendarOutlined />} />}
              title={item.title}
              description={item.detail}
            />
          </List.Item>
        )}
      />
    )
  }
}

```

```

}

export default ConfList

```

- myconf

```

/*
 * @Author: Samrito
 * @Date: 2021-11-19 19:39:36
 * @LastEditors: Samrito
 * @LastEditTime: 2021-11-23 19:34:51
 */
import React, { Component } from 'react'
import { List, Avatar } from 'antd'
import { CalendarOutlined } from '@ant-design/icons';
import 'antd/dist/antd.css'
import { connect } from 'react-redux'
import contract from '../../contracts/contract'

const data = [
  {
    title: 'Ant Design Title 1',
  },
  {
    title: 'Ant Design Title 2',
  },
  {
    title: 'Ant Design Title 3',
  },
  {
    title: 'Ant Design Title 4',
  },
];

class MyConf extends (Component) {
  componentDidMount() {
    //先执行一遍查询操作
    contract.methods.queryMyConf()
      .call({from: window.web3.eth.accounts[0]}), (err, res) => {
        this.setState({loading: true});
        for (var i = 0; i < res.length; i += 1) {
          data.push({title: res[i]});
        }
      }) //function中的res为方法返回值
      .then(() => {
        this.setState({loading: false});
      });

    contract.events.MyNewConference({
      filter: {}, // Using an array means OR: e.g. 20 or 23
      fromBlock: window.web3.eth.getBlockNumber()
    }, (error, event) => {
      this.setState({loading: true});
      data.push({title: event.returnValues[0]});
    }) //重新执行查询操作
  }

  render() {

```

```
    return (
      <List
        itemLayout="horizontal"
        dataSource={data}
        pagination={{ pageSize: 3, simple: 'true' }}
        renderItem={item => (
          <List.Item>
            <List.Item.Meta
              avatar={<Avatar icon={<CalendarOutlined />} />}
              title={item.title}
            />
          </List.Item>
        )}
      />
    )
  }
}

export default MyConf
```