

day90-sol-real-estate-starter2

≡ 태그	
📅 날짜	@ 2023년 2월 3일

vsCODE ganache 커닝고 진행

RealEstate.sol

```
pragma solidity ^0.5.0;

contract RealEstate {
    struct Buyer{
        address buyerAddress;
        string name;
        uint256 age;
    }

    mapping(uint => Buyer) public buyerInfo;

    address payable public owner;
    address[10] public buyers;

    event LogBuyRealEstate(
        address _buyer,
        uint _id
    );

    constructor() public{
        //컨트랙트를 발행한 소유자
        owner = msg.sender;
    }

    function buyRealEstate(uint _id, string memory _name, uint256 _age) public payable{
        require(_id >= 0 && _id <= 9);
        buyers[_id] = msg.sender;
        buyerInfo[_id] = Buyer(msg.sender, _name, _age);

        owner.transfer(msg.value);
        emit LogBuyRealEstate(msg.sender, _id);
    }
}
```

- truffle migrate --network ganache

배포

- truffle console --network ganache

접속

- RealEstate.deployed().then(function(instance) {app = instance})

undefined

- app.LogBuyRealEstate({fromBlock:0, toBlock:"latest"}, function(error, event){console.log(event);})

```
truffle(ganache)> app.LogBuyRealEstate({fromBlock:0, toBlock:"latest", function(error, event){console.log(event);}});
undefined
Subscription {
  _events: Events <[Object: null prototype] {}> {},
  _eventsCount: 0,
  id: null,
  callback: [Function: intermediary],
  arguments: [
    {
      fromBlock: '0x0',
      toBlock: 'latest',
      topics: [Array],
      address: '0x1517efa5f0bf43d4d48b7b81f2347af88e91154c'
    }
  ],
  _reconnectIntervalId: null,
  options: {
    subscription: {
      params: 1,
      inputFormatter: [Array],
      outputFormatter: [Function: bound ],
      subscriptionHandler: [Function: subscriptionHandler]
    },
    type: 'eth',
    requestManager: RequestManager {
      provider: [HttpProvider],
      providers: [Object],
      subscriptions: {}
    }
  },
  subscriptionMethod: 'logs'
}
```

- `app.buyRealEstate(0,"lmh",20,{from:accounts[1], value: web3.utils.toWei("1","ether")})`

buyer 정보

[illegible]

RealEstate.sol

```
pragma solidity ^0.5.0;

contract RealEstate {
    struct Buyer{
        address buyerAddress;
        string name;
        uint256 age;
    }

    mapping(uint => Buyer) public buyerInfo;

    address payable public owner;
    address[10] public buyers;

    event LogBuyRealEstate(
        address _buyer,
        uint _id
    );

    constructor() public{
        //컨트랙트를 발행한 소유자
        owner = msg.sender;
    }

    function buyRealEstate(uint _id, string memory _name, uint256 _age) public payable{
        require(_id >= 0 && _id <= 9);
        buyers[_id] = msg.sender;
        buyerInfo[_id] = Buyer(msg.sender, _name, _age);

        owner.transfer(msg.value);
        emit LogBuyRealEstate(msg.sender, _id);
    }

    function getBuyerInfo(uint _id) public view returns(address, string memory, uint){
        // 구조체에 buyerInfo배열 정보 넣기
        Buyer memory buyer = buyerInfo[_id];

        // key 값에 따른 buyer 정보
        return(buyer.buyerAddress, buyer.name, buyer.age);
    }

    function getAllBuyers() public view returns (address[10] memory){
        return buyers;
    }
}
```

접속되어 있는 상태에서는 truffle만 없애고 명령어를 입력하면 작동한다.

truffle(ganache)> migrate --compile-all --reset

```
truffle(ganache)> migrate --compile-all --reset
> Warning: possible unsupported (undocumented in help) command line option: --compile-all

Compiling your contracts...
=====
> Compiling .\contracts\RealEstate.sol
> Artifacts written to C:\TRUFFLE\real-estate\build\contracts
> Compiled successfully using:
   - solc: 0.5.12+commit.7709ece9.Emscripten.clang

Starting migrations...
=====
> Network name:      'ganache'
> Network id:       5777
> Block gas limit:  0x6691b7
```

```

1_initial_migration.js
=====

Replacing 'Migrations'
-----
> transaction hash: 0x994b6cef8951856a3914795803097ad27215af1cd7d1180a4e2a4e9d37c897f6
> Blocks: 0        Seconds: 0
> contract address: 0x3E19CB39CfC8FCFe93aB22DfBE1AC17294aBEd22
> block number: 8
> block timestamp: 1675386504
> account: 0xa231074A69a9e4eAa36E3A66Fd5CDFE97AB5D7C3
> balance: 102.98050816
> gas used: 225237
> gas price: 20 gwei
> value sent: 0 ETH
> total cost: 0.00450474 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost: 0.00450474 ETH

```

```

2_deploy_contracts.js
=====

Replacing 'RealEstate'
-----
> transaction hash: 0x05309eb7cf89898f46b76cf70f3205320c2a699329c23f7b09169a98a8247749
> Blocks: 0        Seconds: 0
> contract address: 0x83829196F47f535E66c8f8e5Cc061f6423281125
> block number: 10
> block timestamp: 1675386505
> account: 0xa231074A69a9e4eAa36E3A66Fd5CDFE97AB5D7C3
> balance: 102.96669954
> gas used: 648068
> gas price: 20 gwei
> value sent: 0 ETH
> total cost: 0.01296136 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost: 0.01296136 ETH

Summary
=====
> Total deployments: 2
> Final cost: 0.0174661 ETH

truffle(ganache)> 

```

한번더 실행해보기

```

truffle(ganache)> RealEstate.deployed().then(function(instance) {app = instance; })
undefined
truffle(ganache)> app.buyRealEstate(0,"lmh",20,{from:accounts[1], value: web3.utils.toWei("1","ether")})
{

```

buyer 정보 호출 하기

- `app.getBuyerInfo(0);`

```
truffle(ganache)> app.getBuyerInfo(0);
Result {
  '0': '0x1a18BD859B2467E8652EF40b2d1490a99d92E6A7',
  '1': 'lmh',
  '2': BN {
    negative: 0,
    words: [ 20, <1 empty item> ],
    length: 1,
    red: null
  }
}
```

전제 buyers 불러오기

- `app.getAllBuyers();`

[illegible]

TestRealEstate.js

```

var RealEstate = artifacts.require("./RealEstate.sol");

contract('RealEstate', function(accounts){

    var realEstateInstance;

    it("컨트랙트의 소유자 초기화 테스트", function(){
        return RealEstate.deployed().then(function(instance){
            realEstateInstance = instance;
            return realEstateInstance.owner.call();
        }).then(function(owner){
            //대문자
            assert.equal(owner.toUpperCase(), accounts[0].toUpperCase(),
                "owner가 가나슈 첫번째 계정과 동일하지 않습니다.");
        });
    });

    it("가나슈 두번째 계정으로 매물 아이디 0번 매입 후 이벤트 생성 및 매입자 정보와 buyers 배열 테스트", function(){

        return RealEstate.deployed().then(function(instance){
            realEstateInstance = instance;
            return realEstateInstance.buyRealEstate(0, "sbs", 20, {from: accounts[1], value: web3.utils.toWei("1", "ether")}));
        }).then(function(receipt){
            assert.equal(receipt.logs.length, 1, "이벤트 하나가 생성되지 않았습니다.");
            assert.equal(receipt.logs[0].event, "LogBuyRealEstate", "이벤트가 LogBuyRealEstate가 아닙니다.");
            assert.equal(receipt.logs[0].args._buyer, accounts[1], "매입자가 가나슈 두번째 계정이 아닙니다.");
            assert.equal(receipt.logs[0].args._id, 0, "매물 아이디가 0이 아닙니다.");

            return realEstateInstance.getBuyerInfo(0);
        }).then(function(buyerInfo){
            assert.equal(buyerInfo[0].toUpperCase(), accounts[1].toUpperCase(),
                "매입자의 계정이 가나슈 두번째 계정과 일치하지 않습니다.");
            assert.equal(web3.utils.toAscii(buyerInfo[1]).replace(/\0/g, ''), "aaa", "매입자의 이름이 aaa가 아닙니다.");
        });
    });
});

```

```

        assert.equal(buyerInfo[2], 20, "매입자의 나이가 20살이 아닙니다.");
        return realEstateInstance.getAllBuyers();
    }).then(function(buyers){
        assert.equal(buyer[0].toUpperCase(), accounts[1].toUpperCase(), "Buyers 배열 첫번째 인덱스의 계정이 가나슈 두번째 계정과 일치x");
    })
});

```

- `RealEstate.deployed().then(function(instance) {app = instance; })`

undefined

- `app.buyRealEstate(1,"lmh",20,{from:accounts[1], value: web3.utils.toWei("1","ether")})`

[illegible]

- `app.getAllBuyers();`

```

}
truffle(ganache)> app.getAllBuyers();
[
  '0x1a1BBD859B2467E8652EF40b2d1490a99d92E6A7',
  '0x1a1BBD859B2467E8652EF40b2d1490a99d92E6A7',
  '0x0000000000000000000000000000000000000000',
  '0x0000000000000000000000000000000000000000',
  '0x0000000000000000000000000000000000000000',
  '0x0000000000000000000000000000000000000000',
  '0x0000000000000000000000000000000000000000',
  '0x0000000000000000000000000000000000000000',
  '0x0000000000000000000000000000000000000000',
  '0x0000000000000000000000000000000000000000',
  '0x0000000000000000000000000000000000000000',
  '0x0000000000000000000000000000000000000000'
]
truffle(ganache)> 

```

- truffle test --network ganache

TestRealEstate.js

```

var RealEstate = artifacts.require("./RealEstate.sol");

contract('RealEstate',function(accounts){
  var realEstateInstance;

  it("컨트랙트의 소유자 초기화 테스트",function(){
    return RealEstate.deployed().then(function(instance){
      realEstateInstance = instance;
      return realEstateInstance.owner.call();
    }).then(function(owner){
      //대문자
      assert.equal(owner.toUpperCase(),accounts[0].toUpperCase(),
        "owner가 가나슈 첫번째 계정과 동일하지 않습니다.");
    });
  });

  it("가나슈 두번째 계정으로 매물 아이디 0번 매입 후 이벤트 생성 및 매입자 정보와 buyers 배열 테스트",function(){

    return RealEstate.deployed().then(function(instance){
      realEstateInstance = instance;
      return realEstateInstance.buyRealEstate(0,"aaa",20,{from: accounts[1],value:
        web3.utils.toWei("1","ether")}));
    }).then(function(receipt){
      assert.equal(receipt.logs.length,1,"이벤트 하나가 생성되지 않았습니다.");
      assert.equal(receipt.logs[0].event,"LogBuyRealEstate","이벤트가 LogBuyRealEstate가 아닙니다.");
      // assert.equal(receipt.logs[0].args._buyer,accounts[1],"매입자가 가나슈 두번째 계정이 아닙니다.");
      // assert.equal(receipt.logs[0].args._id,0,"매물 아이디가 0이 아닙니다.");
      return realEstateInstance.getBuyerInfo(0);
    }).then(function(buyerInfo){
      // assert.equal(buyerInfo[0].toUpperCase(),accounts[1].toUpperCase(),
      //   "매입자의 계정이 가나슈 두번째 계정과 일치하지 않습니다.");
      // assert.equal(web3.utils.toAscii(buyerInfo[1]).replace(/\0/g,''),'aaa',"매입자의 이름이 aaa가 아닙니다.");
      assert.equal(buyerInfo[2],20,"매입자의 나이가 20살이 아닙니다.");
      return realEstateInstance.getAllBuyers();
    }).then(function(buyers){
      // assert.equal(buyers[0].toUpperCase(),accounts[1].toUpperCase(),"Buyers 배열 첫번째 인덱스의 계정이 가나슈 두번째 계정과 일치x");
    })
  });
});

```

위 명령어로 실행 하여 에러가 발생하는 부분을 하나씩 주석 처리하면서 패싱이 되는지 확인한다.

- npm install

(lite-server등 자동으로 설치 되게)

B

문서 예 아이콘 테마 블로그

Q 찾다

flexbox 가이드를 읽어보세요.

열

HTML

```

<div class="container text-center">
  <div class="row">
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
  </div>
</div>

```

위의 예는 사전 정의된 그리드 클래스를 사용하여 모든 것을 쉽게 만들 수 있습니다. 이러한 열은 부모가 있는 페이지 중앙에 배치됩니다.

작동 방식

그리드 시스템이 결합되는 방식은 다음과 같습니다.

- 당사의 그리드는 **6개의 반응형 중단점**을 지원합니다. 즉, 해당 중단점과 그 위에 있는 모든 항목에 영향을 미칠 수 있는 컨테이너 및 열 크기와 동작을 제어할 수 있습니다.
- 컨테이너는 콘텐츠를 중앙에 놓고 가로로 채웁니다. 비례 사용 하거나 유체 및 픽셀 너비의 조합에 대한

시작하기

소개

다운로드

내용물

브라우저 및 기기

자바스크립트

웹팩

소포

초대

접근성

RFS

RTL

기여하다

커스터마이징

개요

사스

옵션

색상

색상 모드

구성품

CSS 변수

최적화

공들여 나열한 것

중단점

컨테이너

그리드

열

흐름

```

<!-- Bootstrap -->
<link href="css/bootstrap.min.css" rel="stylesheet">
<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
<![endif]-->
</head>
<body>
  <div class="container">

```

이렇게 부트스트랩 코드를 넣고

day90-sol-real-estate-starter2

8


```

</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col-xs-12 col-sm-8 col-sm-push-2">
        <h1 class="text-center">이더리움 부동산</h1>
        <hr/>
        <br/>
      </div>
    </div>

    <div class="row" id="list">
      <!-- 매물 리스트 -->
    </div>

  </div>

  <div id = "template" style="display : none;">
    <div class="col-sm-6 col-md-4 col-lg-3">

```

사이트에서도 확인 가능하고 위 처럼 작성하게 되면 프론트를 꾸밀 수 있다.

index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after* these tags -->
    <title>이더리움 부동산</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-xs-12 col-sm-8 col-sm-push-2">
          <h1 class="text-center">이더리움 부동산</h1>
          <hr/>
          <br/>
        </div>
      </div>

      <div class="row" id="list">
        <!-- 매물 리스트 -->
      </div>
    </div>

    <div id = "template" style="display : none;">
      <div class="col-sm-6 col-md-4 col-lg-3">
        <div class="panel panel-success panel-realEstate">
          <div class="panel-heading">
            <h3 class="panel-title"> > 매물</h3>
          </div>
          <div class="panel-body">
            <img style="width : 100%;" src="">
            <br/><br/>
            <strong>아이디</strong> : <span class="id"></span> <br/>
            <strong>종류</strong> : <span class="type"></span> <br/>
            <strong>면적(m2)</strong> : <span class="area"></span> <br/>
            <strong>가격(ETH)</strong> : <span class="price"></span> <br/>
            <button class="btn btn-info btn-buy">

```

```

        type="button"
        data-toggle="modal"
        data-target="#buyModal">
        매입
    </button>
    <button class="btn btn-info btn-buy"
        type="button"
        data-toggle="modal"
        data-target="#buyerInfoModal"
        style="display : none;">
        매입자 정보
    </button>
</div>
</div>
</div>

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<!-- Include all compiled plugins (below), or include individual files as needed -->
<script src="js/bootstrap.min.js"></script>
<script src="js/web3.min.js"></script>
<script src="js/truffle-contract.js"></script>
<script src="js/app.js"></script>
<script src="js/utf8.js"></script>
</body>
</html>

```

app.js

```

App = {
  web3Provider: null,
  contracts: {},

  init: function() {
    //제이쿼리 제이슨 가져오기
    $.getJSON('../real-estate.json',function(data){
      // 여기에 뿌리게 된다.
      // <div class="row" id="list">
      //   <!-- 매물 리스트 -->
      // </div>

      var list = $('#list');

      // <div id = "template" style="display : none;">
      // 위 부분을 가져온다.
      // 템플릿 아이디
      var template = $('#template');

      for (i = 0; i < data.length; i++){
        //템플릿에 이미지 태그에 src 그림파일 위치 넣어주기
        template.find('img').attr('src', data[i].picture);
        template.find('.id').text(data[i].id);
        template.find('.type').text(data[i].type);
        template.find('.area').text(data[i].area);
        template.find('.price').text(data[i].price);

        list.append(template.html());
      }
    })
  },

  initWeb3: function() {

  },

  initContract: function() {

  },

  buyRealEstate: function() {

  },

  loadRealEstates: function() {

  },

  listenToEvents: function() {

```

```
    }  
  };  
  
  $(function() {  
    $(window).load(function() {  
      App.init();  
    });  
  });  
});
```

프론트 확인 해보기

- npm run dev