

Document of Ethereum Project

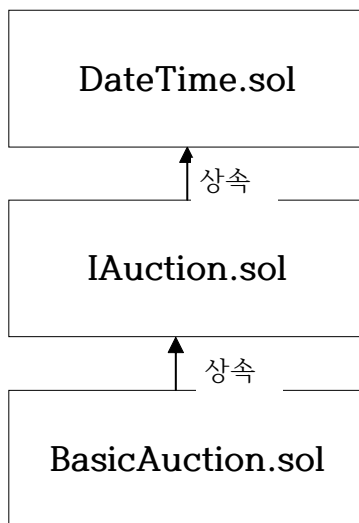
2015004302 곽상원

목차

1. Smart contract
 2. Web3.js
 3. 실행 및 실행결과
-

1. Smart contract

1)구조



2)Source code

①DateTime.sol:

uint형 time data를 문자열형태로 표현하기 위해,

[‘https://github.com/pipermerriam/ethereum-datetime’](https://github.com/pipermerriam/ethereum-datetime)를 참조했다.

②IAuction.sol:

경매관련 contract의 struct, event, function이 정의 및 선언됐다.

▶ Struct

-RegisteredItem: 경매장에 등록된 물품을 위한 구조체이다.

fill: 해당 이름의 물품이 등록돼있으면 true 아니면 false, item: 등록된 물품명,
owner: 물품을 등록한 사람의 address, cur_price: 현재 입찰 가격
cur_bidder: 현재 입찰자, upper_limit_price: 상한가, due_date: 경매가 끝나는 시각

-AuctionedItem: 경매가 마감된 물품을 위한 구조체이다.

fill: 해당 이름의 물품이 경매 마감됐으면 true 아닌 경우에 false를 갖는다.
item: 마감된 물품명, owner: 마감된 물품의 소유주, winning_bid: 낙찰가격

▶ Event

-Register_name(): 이름을 등록하면 발생

-Register_item(): 물품을 개인창고에 등록하면 발생

-Auction_bidding(): 입찰시에 발생

-Transfer(): 송금시에 발생

-Due_over(): 등록된 물품의 기간이 지나면 발생

-Auction_end(): 해당 물품에 대한 경매가 종료시 발생

▶function: 이에 대한 내용은 아래의 BasicAuction.sol에서 다룬다.

③ BasicAuction.sol:

경매장 contract를 실제 구현한 코드로 핵심적인 전역변수, function이 된다.

▶전역변수

-_myItem: 개인창고에 해당

-IM: address를 key로, user name을 value로 갖는 mapping

-RItem: 물품명을 key로, 그에 해당하는 경매에 등록된 물품을 value로 갖는 mapping

-RItem_list: 경매에 등록된 물품명을 갖는 배열로, RItem과 함께 쓰임으로써 등록된 모든 물품을 확인하는 용도이다.

-AItem: 물품명을 key로, 그에 해당하는 마감된 물품을 value로 갖는 mapping

-AItem_list: 마감된 물품명을 갖는 배열로, AItem과 함께 쓰임으로써 마감된 모든 물품을 확인하는 용도이다.

► Function

-registerName(name)

msg.sender의 user name을 'name'으로 바꿀 때 사용한다.
IM에 msg.sender => name으로 저장하고 Register_name() event를 emit.

-registerItem(name)

개인창고에 해당 name으로 물품을 등록할 때 사용한다.
먼저 require문을 이용해 경매장에 이미 경매에 등록된 물품이나 경매가 끝난 물품과 이름이 중복되지 않도록 체크한다. 조건이 맞으면 _myItem에 해당 name으로 물품을 등록하고 Register_item() event를 emit.

-_registerItem(name)

위의 registerItem()과 기능은 같지만 require문을 이용한 name 중복 체크를 하지 않는다.
이는 auctionEnd()에서 경매가 끝난 물품의 소유주를 바꿀 때 사용된다.

-registerAuctionItem(name, start_price, limit_price, date)

개인창고에 있는 name 물품을 시작가 start_price, 상한가 limit_price, 현 시각으로부터 date후에 경매를 마감하도록 등록하는데 사용한다. 이때 date는 minute 단위를 이용한다.
먼저 require문을 이용해 시작가보다 상한가가 높은지를 체크한다. 올바르면 RItem에 해당 물품정보를 저장함으로써 경매장에 등록한다. 이때 storage형으로 변수를 선언해서 EVM의 storage에 저장할 수 있도록 했고, 현재 입찰자를 address(0)로 함으로써 아직 아무도 입찰하지 않았음을 나타낸다. 마지막으로 RItem_list에 name을 추가한다.

-auctionBidding(name, value)

기본적으로 경매장에 등록된 물품 name에 value만큼 입찰하는 용도로, value에 해당하는 ether만큼 경매장측에 송금한다.
먼저 if문을 이용해 물품의 마감일이 지났는지 확인한다.
만약 마감일이 지났다면 현재 입찰자는 value에 해당하는 금액을 payback받고, auctionEnd()를 호출해 이 물품에 대한 경매를 종료하고 Due_over() event를 emit.
마감일이 지나지 않았으면, require문으로 현재 물품에 입찰된 가격보다 value가 더 큰지 체크한다. 올바르면, 이전 입찰자가 이전에 입찰한 가격만큼 payback받을 필요가 있다. 이때 address(0)이면 아직 아무도 입찰하지 않은 상태이므로 payback할 필요가 없다.
그리고나서 현재 입찰가와 입찰자를 갱신하고 Auction_bidding event를 emit.
마지막으로 현재 입찰가가 상한가보다 높으면 그 차액만큼을 payback하고 이 물품에 대한 경매를 종료한다.

-auctionEnd(item)

경매장에 등록된 물품 item에 대한 경매가 종료될 때 호출된다.

경매가 종료됐을 때 2가지 경우가 생길 수 있는데 이를 if문으로 걸러낸다.

1. 누군가 입찰한 경우: 물품을 등록한 사람에게 balanceTransfer()를 이용해 낙찰가만큼 송금한다. 그리고 changeItemOwner()를 이용해 물품에 대한 소유주를 바꾸고 등록된 물품에서 경매가 끝난 물품으로 전환해준다.
2. 아무도 입찰하지 않은 경우: 경매에 등록된 물품에서 경매가 끝난 물품으로만 바꿔주면 된다,

마지막으로 Auction_end() event를 emit.

-balanceTransfer(seller, price): seller에게 price에 해당하는 ether만큼 전송한다.

-changeItemOwner(item)

경매장에 등록된 item에 대한 소유주를 바꾸고, 등록된 물품에서 경매가 끝난 물품으로 전환해준다.

item에 대한 소유주를 바꾸기 위해, 물품 등록자의 개인창고인 _myItem[item.owner]에서 해당 물품의 이름을 지운다. 그리고 _registerItem()을 이용해 낙찰자의 개인창고에 item을 등록한다. 마지막으로 RegToAuc()을 이용해 등록된 물품에서 경매가 끝난 물품으로 전환해준다.

-RegToAuc(item, who)

낙찰자인 who를 새로운 owner로서, item을 등록된 물품에서 경매가 끝난 물품으로 바꿔준다.

AuctionedItem 구조체의 내용을 알맞게 채우고 Altem_list에 item을 추가해준다. 그리고 RItem_list에서 item을 제거해준다.

-getMyItems(): 개인창고에 있는 물품명을 string[]형으로 반환한다.

-getName(): msg.sender의 user name을 string형으로 반환한다.

-getName_2(address owner): owner에 해당하는 user name을 string형으로 반환한다.

-getAllRegisteredItems(): 경매에 등록된 모든 물품을 RegisteredItem[]형으로 반환한다.

-getAllAuctionedItems(): 경매가 끝난 모든 물품을 AuctionedItem[]형으로 반환한다.

-getTranstedTime(time)

now keyword를 이용해 얻은 시각이 대한민국 시각과 9시간만큼 차이가 나므로 이를 보완하기 위한 함수이다.

이 함수는 Dapp에서 물품의 마감시간을 대한민국 시각을 기준으로 문자열형태로 표현하기 위해 사용된다.

2.Web3.js

▶환경

index.ejs를 html파일로 사용한다.

Localhost의 8545번 포트에서 private ethereum network와 연결된다.

▶Function

-*startDapp()*: 경매장 app이 실행됐을 때 호출할 함수를 담고 있다.

-*getBalance()*: User address에 해당하는 ether 잔액을 보여주기 위해 사용된다.

-*getName()*

Smart contract의 getName()을 호출할 때 msg.sender를 user address로 설정해 user address에 맵핑되는 name을 가져오고 이를 id("name")의 html에 반영한다.

-*registerName()*

Id("change_name")에 해당하는 값을 가져와 smart contract의 registerName()을 이용해 user address에 맵핑되는 name을 바꾼다. 그리고 getName()으로 html에 결과를 반영한다.

-*registerForMyItem()*

Id("Item")에 해당하는 값을 가져와 smart contract의 registerItem()을 이용해 user address의 개인창고에 물품을 추가한다. 그리고 getMyItems()로 html에 결과를 반영한다.

-*registerAuctionItem()*:

개인창고에 있는 물품을 경매장에 등록하기위한 함수로 smart contract의 registerAuctionItem()을 호출해 물품을 등록한다. 이때 마감시간인 due date는 minute단위이므로 minute, hour, day, week에 대해 적절한 값을 곱했다. 마지막으로 getItemsRegisteredAtAuction()과 getRegisteredAuctionItems()를 호출해 html에 결과를 반영했다.

-*auctionBidding()*:

경매장에 등록된 물품에 대해 입찰을 할 때 사용하는 함수로 smart contract의 auctionBidding()을 이용한다. 그리고 getRegisteredAuctionItems()와 getClosedAuctionItems()를 호출해 html에 그 결과를 반영한다.

-getRegisteredAuctionItems():

Smart contract의 getAllRegisteredItems()를 호출해 경매장에 등록된 모든 물품목록을 가져온다. 그리고 마감시각을 string으로 표현할 수 있도록 smart contract의 getTranstedTime()을 호출했다. 마지막으로 그 결과를 html에 반영한다.

-getClosedAuctionItems():

Smart contract의 getAllAuctionedItems()를 호출해 경매가 끝난 모든 물품목록을 가져온다. 그리고 smart contract의 getName_2()를 이용해 물품의 소유주를 user name으로 표현한다. 마지막으로 그 결과를 html에 반영한다.

-getMyItemsToBeAuctioned():

개인창고의 목록을 가져와 경매에 등록할 물품을 보여주기 위한 함수로, smart contract의 getMyItems()를 이용해 목록을 가져오고 그 결과를 html에 반영한다.

-getItemsRegisteredAtAuction():

경매에 등록된 물품목록을 가져와 입찰하려는 경우에 그 목록을 보여주기위한 함수로, smart contract의 getAllRegisteredItems()를 이용해 목록을 가져오고 그 결과를 html에 반영한다.

3. 실행 및 실행결과

▶시나리오

- ①A가 “myTem”을 개인창고에 등록한다.->②A가 “myTem”을 경매장에 등록한다.->
 - ③B가 상한가미만의 적절한 가격으로 입찰한다.->④마감시간이 지나 B가 낙찰자가 된다.->
 - ⑤잔액확인
-

①A가 “myTem”을 개인창고에 등록한다.

1)A가 “myTem”을 등록

The screenshot shows a web browser window with two tabs: "Remix - Ethereum IDE" and "경매 시스템". The address bar shows "localhost:3000/change_name=A#". The page content is divided into two main sections. The top section, titled "Identity", displays user information: "Key : 0x86BCdCDB2669F00df35d802f4b279C401652B4", "Ether : 99.94628694", and "Name : A". Below this are two form groups. The first, "change name", has a text input field with the placeholder "Enter user name" and a red "CHANGE NAME" button. The second, "change user", has a text input field with the placeholder "Enter ethereum key" and a red "CHANGE USER" button. The bottom section is split into two columns. The left column, titled "Register for My Item", contains a form with a text input field labeled "Item" containing the text "myTem", and two red buttons labeled "REGISTER" and "RESET". The right column, titled "My Items", contains a text input field labeled "Item". Below these columns are two more section titles: "Register Auction Item" and "Auction bidding".

2)A의 개인창고에 “myTem”이 추가됨

This screenshot shows the same web application interface as the previous one, but with the URL changed to "localhost:3000/?item=myTem#". In the "My Items" section, the text input field now displays "myTem", indicating that the item has been successfully registered to the user's personal storage.

②A가 “myTem”을 경매장에 등록한다.

1)A가 “myTem”을 시작가 0, 상한가 20, 마감시간까지 5분으로 경매장에 등록
(due date를 입력할 때 아래의 선택창에서 minute, hour, day, week 단위를 선택할 수 있음)

The screenshot shows a web application interface for an auction system. The browser tabs include 'Remix - Ethereum IDE' and '경매 시스템'. The address bar shows 'localhost:3000/item=myTem#'. The interface is divided into two main columns.

Left Column:

- Register for My Item:** Contains a form with an 'Item' input field (containing 'myTem'), a 'REGISTER' button, and a 'RESET' button.
- Register Auction Item:** Contains a dropdown menu (selected 'myTem'), 'starting bid price' (0), 'upper limit price' (20), 'due date' (Minute, 5), and 'REGISTER'/'RESET' buttons.
- Registered auction items:** A table with columns: Item, Owner, Bid price, Upper limit price, Due date.

Right Column:

- My Items:** Contains an 'Item' input field (containing 'myTem').
- Auction bidding:** Contains a dropdown menu (selected 'myTem'), a 'bid price' input field (containing 'price'), and 'BID'/'RESET' buttons.
- Closed auction items:** A table with columns: Item, Owner, Winning bid.

2) “myTem”이 경매장 물품에 추가됨

This screenshot shows the same interface as the previous one, but with the 'Registered auction items' table populated with data for 'myTem'.

Registered auction items:

Item	Owner	Bid price	Upper limit price	Due date
myTem	A	0	20	2020/12.2/21:33

③B가 상한가미만의 적절한 가격으로 입찰한다.

1)B가 “myTem”에 11eth로 입찰

Remix - Ethereum IDE x 경제 시스템 x +

localhost:3000/?change_name=B#

Key : 0xFadeF122697032Bdb3542526A32a7eBBc3662E78

Ether : 99.99909844

Name : B

change name

Enter user name

CHANGE NAME

change user

Enter ethereum key

CHANGE USER

Register for My Item

Item

Item

REGISTER RESET

My Items

Item

Item

Register Auction Item

starting bid price : upper limit price :

price price

Auction bidding

myTem

bid price :

11

2)B의 잔액이 11eth만큼 줄어든 모습

(입찰과 동시에 잔액이 줄어들고, 다른 입찰자가 더 높은 가격을 제시하면 이전 입찰자는 돈을 돌려받는다.)

Remix - Ethereum IDE x 경제 시스템 x +

localhost:3000/?demo-category=myTem&bidPrice=11#

Key : 0xFadeF122697032Bdb3542526A32a7eBBc3662E78

Ether : 88.99770286

Name : B

change name

Enter user name

CHANGE NAME

change user

Enter ethereum key

CHANGE USER

Register for My Item

Item

Item

REGISTER RESET

My Items

Item

Item

Register Auction Item

Auction bidding

3)Registered auction items에 입찰한 결과가 반영됨

Item

Item

REGISTER RESET

REGISTER Auction Item

starting bid price : upper limit price :

price price

due date :

Minute date

REGISTER RESET

Auction bidding

myTem

bid price :

price

BID RESET

Registered auction items

Item	Owner	Bid price	Upper limit price	Due date
myTem	A	11	20	2020/12.2/21:33

Closed auction items

Item	Owner	Winning bid
------	-------	-------------

④마감시간이 지나 B가 낙찰자가 된다.

1) “myTem”의 마감시각이 지나고 임의의 값으로 입찰한다.(이는 contract가 스스로 호출할 수 없기에, 상한가로 물품을 얻은 낙찰자가 아니라면 반드시 마감시각이 지나고 해당 물품에 대해 1회의 재입찰을 해주어야한다.)

Item

Item

REGISTER RESET

REGISTER Auction Item

starting bid price : upper limit price :

price price

due date :

Minute date

REGISTER RESET

Auction bidding

myTem

bid price :

0

BID RESET

Registered auction items

Item	Owner	Bid price	Upper limit price	Due date
myTem	A	11	20	2020/12.2/21:33

Closed auction items

Item	Owner	Winning bid
------	-------	-------------

2) Closed auction items에 “myTem”의 낙찰결과가 표시됨.

The screenshot shows a web application interface for an auction system. The interface is divided into two main columns. The left column contains three sections: "Register for My Item" with an "Item" input field and "REGISTER" and "RESET" buttons; "Register Auction Item" with a dropdown menu set to "myTem", "starting bid price" and "upper limit price" input fields, a "due date" dropdown, and "REGISTER" and "RESET" buttons; and a table titled "Registered auction items" with columns "Item", "Owner", "Bid price", "Upper limit price", and "Due date". The right column contains two sections: "My Items" with an "Item" input field showing "myTem"; and "Auction bidding" with a "bid price" input field and "BID" and "RESET" buttons. Below this is a table titled "Closed auction items" with columns "Item", "Owner", and "Winning bid", showing "myTem" as the item, "B" as the owner, and "11" as the winning bid.

⑤잔액결과

1)A의 잔액이 11eth만큼 증가

Identity

Key : 0x86BCdCBDB2669Ff00df35dB02f4b279C401652B4

Ether : 110.94148402

Name : A

change name

CHANGE NAME

change user

CHANGE USER

Register for My Item

Item

REGISTER

RESET

My Items

Item

2)B의 잔액이 11eth만큼 감소(사실상 입찰하는 타이밍에 이미 감소됐음)

Identity

Key : 0xFadeF122697032Bdb3542526A32a7eBBc3662E78

Ether : 88.99469772

Name : B

change name

CHANGE NAME

change user

CHANGE USER

Register for My Item

Item

REGISTER

RESET

My Items

Item

myTem