# Project writeup

## Background

In this project a token called Tom Property Listing TPL to represent owner title to the properties. Before a token is minted, owner need to verify his ownership to his property. zk-SNARKs is used to create a verification system which can prove owner has title to the property without revealing that specific information on the property.

Once the token has been verified you will place it on a blockchain market place (OpenSea) for others to purchase.

## Technical specification

### Version

Truffle v5.1.21 (core: 5.1.21)

Solidity - 0.5.2 (solc-js)

Node v9.4.0

Web3.js v1.2.1

Contract Address: [0x47FB7b2E4Dfb78A90306B2feCBF16BaF0c78d7Ea](https://rinkeby.etherscan.io/address/0x47FB7b2E4Dfb78A90306B2feCBF16BaF0c78d7Ea#tokentxnsErc721)

### Setup development environment

Code: <https://github.com/ymlai87416/blockchain-nanodegree-capstone>

#### Install

This repository contains Smart Contract code in Solidity (using Truffle), tests (also using Truffle), webapp (using HTML, CSS and JS)

To install, download or clone the repo, then:

npm install

cd eth-contracts

truffle compile

#### Develop Client

Ganache is required to click start the project.

To run Ganache

ganache-cli -m "candy maple cake sugar pudding cream honey rich smooth crumble sweet treat" -a 50 -l 9999999 -q

To run truffle tests:

truffle test ./test/TestERC721Mintable.js

truffle test ./test/TestSolnSquareVerifier.js

truffle test ./test/TestSquareVerifier.js

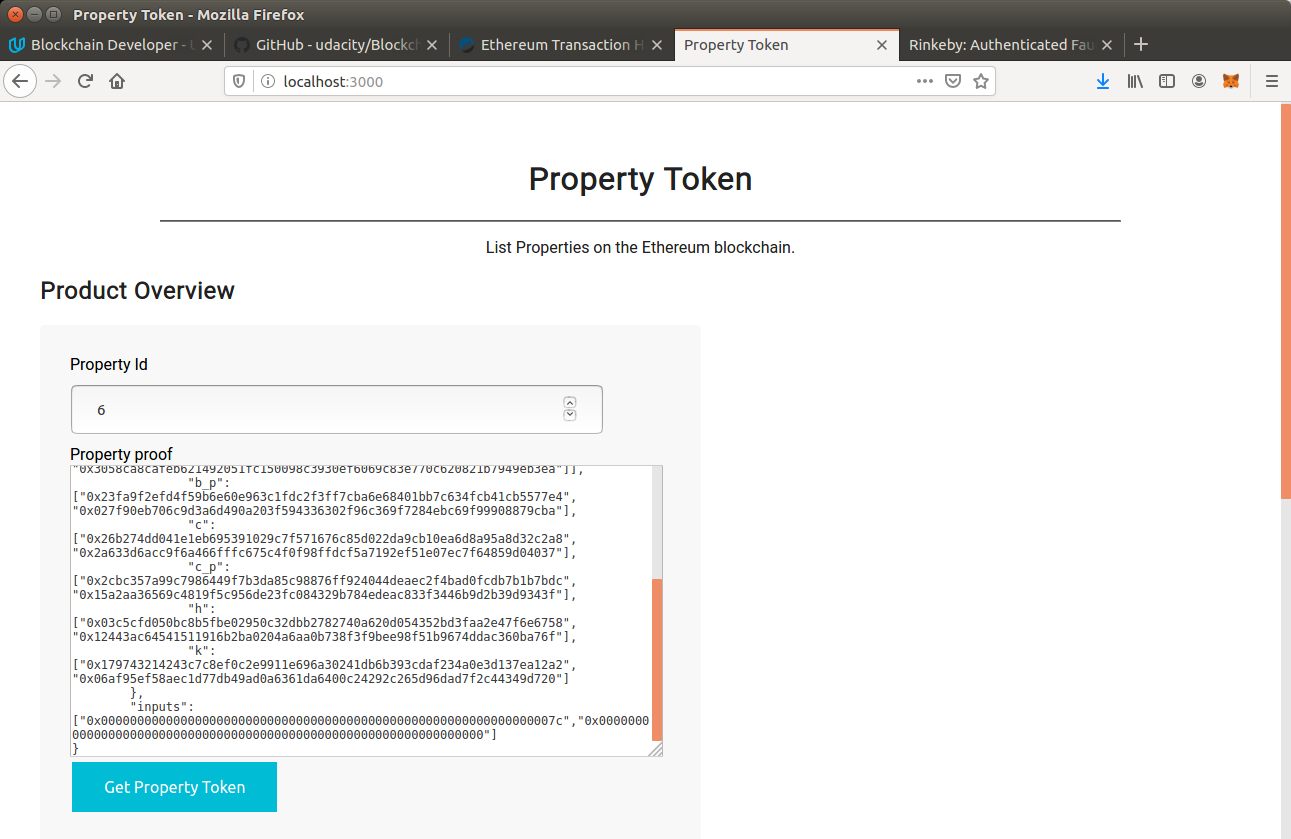
To use the webapp

truffle migrate --reset --network development

npm run dev

To view dapp:

http://localhost:3000



## Project specification

ERC721 is implemented in eth-contracts\contracts\ERC721Mintable.sol

The test code is at: eth-contracts\test\TestERC721Mintable.js

Zokrates program which test user ability to calculate square of a number

DSL code: zokrates\code\square\square.code

Verifier contract: zokrates\code\square\verifier.sol

Proof example: User proof that he knows how to calculate square, and give the answer of 32 = 9

{

"proof": {

"a": ["0x1e738b5a44c64d7b772ccc2638496b61ce2e852a95b4ec05246a492f0bab3328", "0x0260c2177894ced90fc0e8ba79170fc0931c07f7ce777af678abd19132e9bb51"],

"a\_p": ["0x076ff328f180b1e0836b13292a484908ea0708dff5b67e8b9da93081d060d6b4", "0x12eab1a35b78d57be773f508653d6cebb655cfeecbd0f66167b768118f087548"],

"b": [["0x1564b152ed44ae5fe0ff3c6a4666ddfc82a74cb1b1c2695bb0f7519f18d27c2b", "0x26500b7902b193c5d6563ef82671b2d1a64598fcc74e37ac0422c3f0ba42850a"], ["0x1ef319e08716bbcd5c54cc3a4c441493e1f01311879cfc7ab14169769adb8a7f", "0x18cf7d855a38cacd2ffd9089fa5abb64361d9457cf3e57e807ca728f41cbf35f"]],

"b\_p": ["0x09acf2c8afbc862a53af9e33c56ff411b40aa8d4d68491d7bb78c2d15bfeb57a", "0x06dde9b2e64642ca6563772a9180eabaad94cc2cfc00d4a7309b0c1ac7fd5ff7"],

"c": ["0x1857b53c85560029c4bed676d75ee9867fb981b35193503f39323e1f4d3f231e", "0x1565ab0ecbf9369484a944e7ad9998a4f53b5fb7d7d4247a5398f8e142da08ad"],

"c\_p": ["0x2026b17ff8900e55e4ef72321a733effeb56abbff35ba482c3aedec8e4336dbd", "0x29ca5e93cff02132c9e78cac9aa69754a341657223bd44800f9e089a6fadf62e"],

"h": ["0x23fb5b0fcf2ef8d11c655dd7f48bd17ceda9734f85481cabe897ee8839b596fe", "0x16405a652c433ca5a5f59a97aea4e3e6ac3fe5898fc39fa0a6f6371176329726"],

"k": ["0x0e434812f972a2f3eeaeea21a8fb78d4cecfc823aa66d82b8d8b9228de9c7bdb", "0x2753b6627019c510fd6dfe9fe31dfc2d6a10cd44e172b886c6ccbdf94e0596e9"]

},

"inputs": ["0x0000000000000000000000000000000000000000000000000000000000000009","0x0000000000000000000000000000000000000000000000000000000000000001"]

}

The final contract SolnSquareVerifier is an ERC721 token contract, and only mint ERC721 tokens upon user proof that they can calculate the square of a number.

Code: eth-contracts\contracts\SolnSquareVerifiier.sol

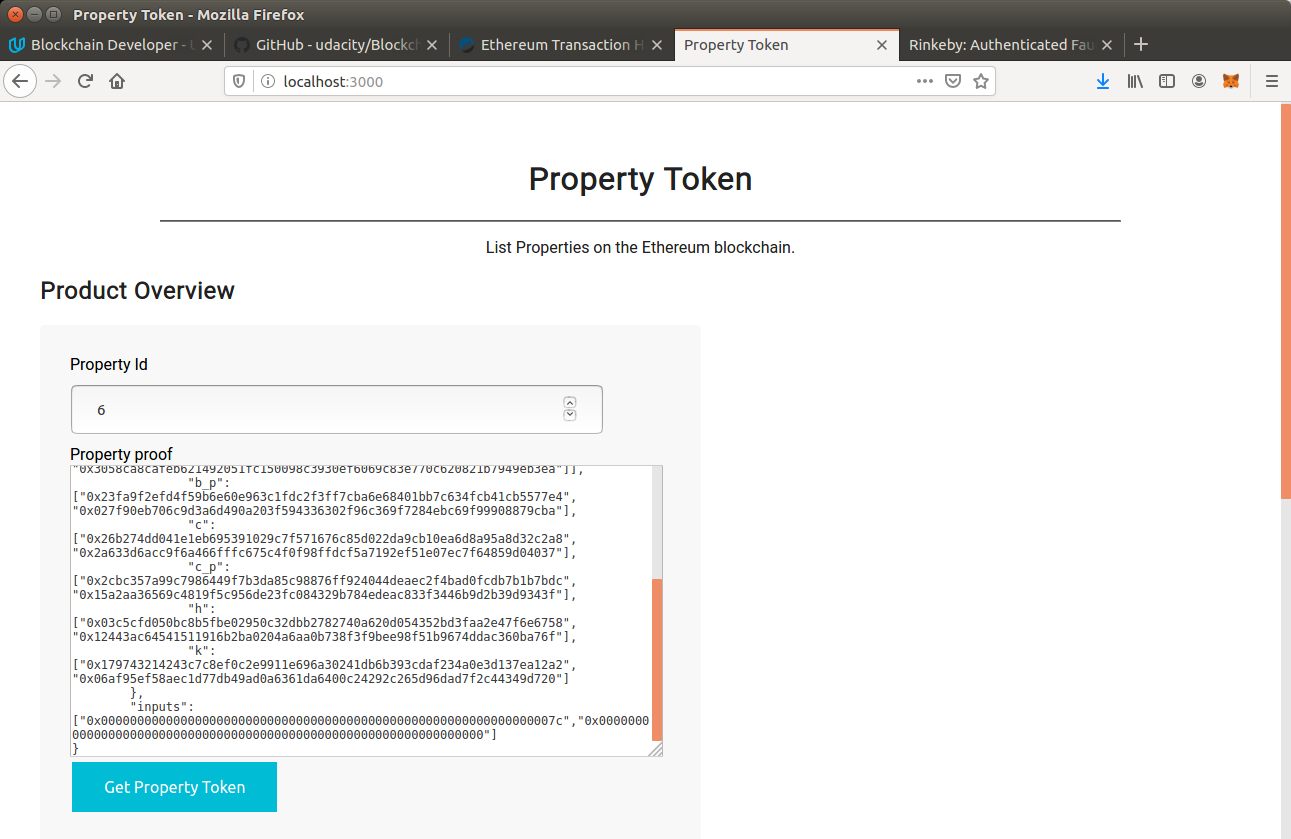
Test:

* eth-contracts\test\TestSquareVerifier.js
* eth-contracts\test\TestSolnSquareVerifier.js

The contract is deployed on Rinkeby network at address: 0x47FB7b2E4Dfb78A90306B2feCBF16BaF0c78d7Ea

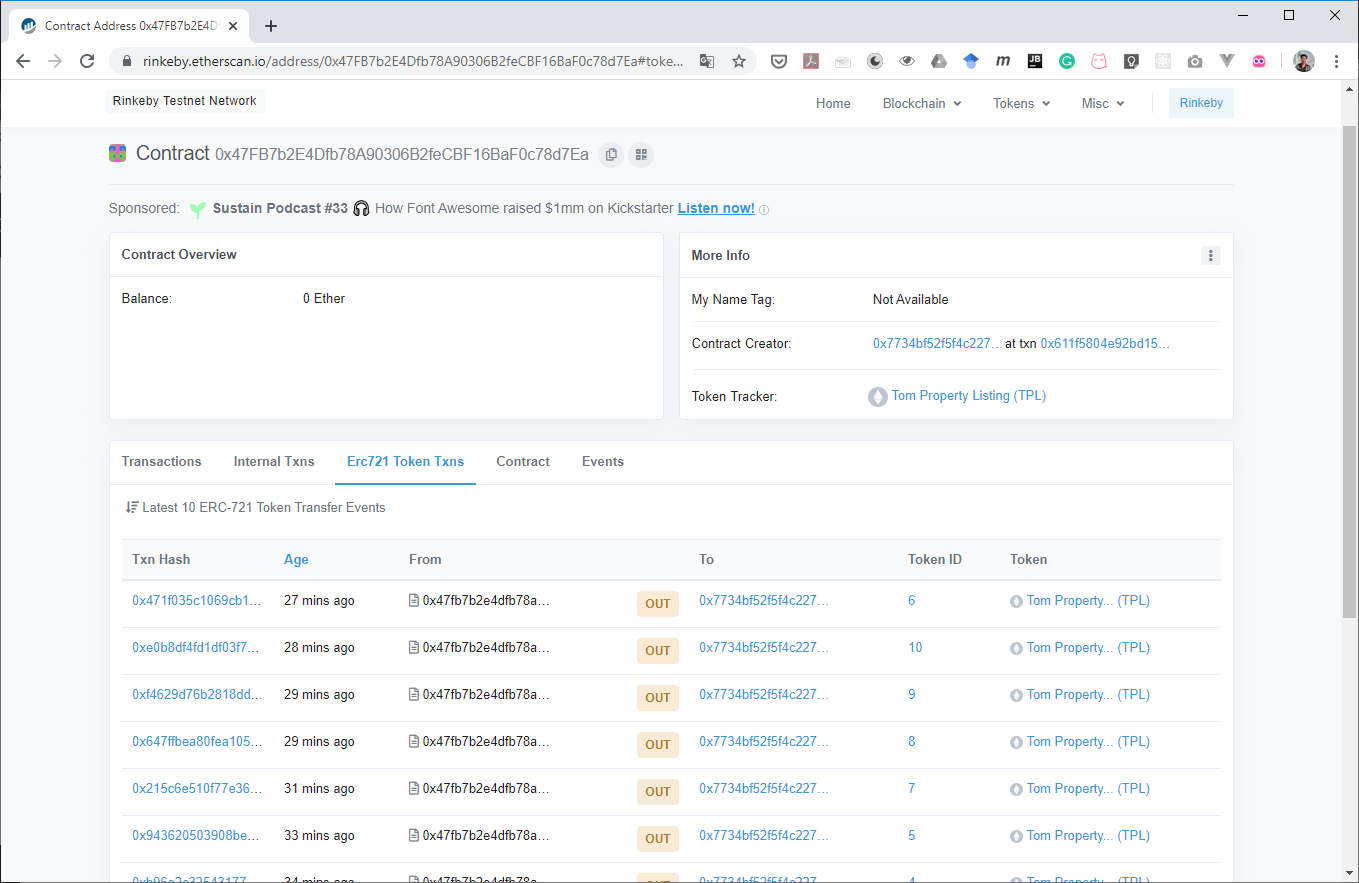
Using the webapp, 10 tokens are minted.

In the webapp, one has to provide the token id and the proof in order to mint a token.

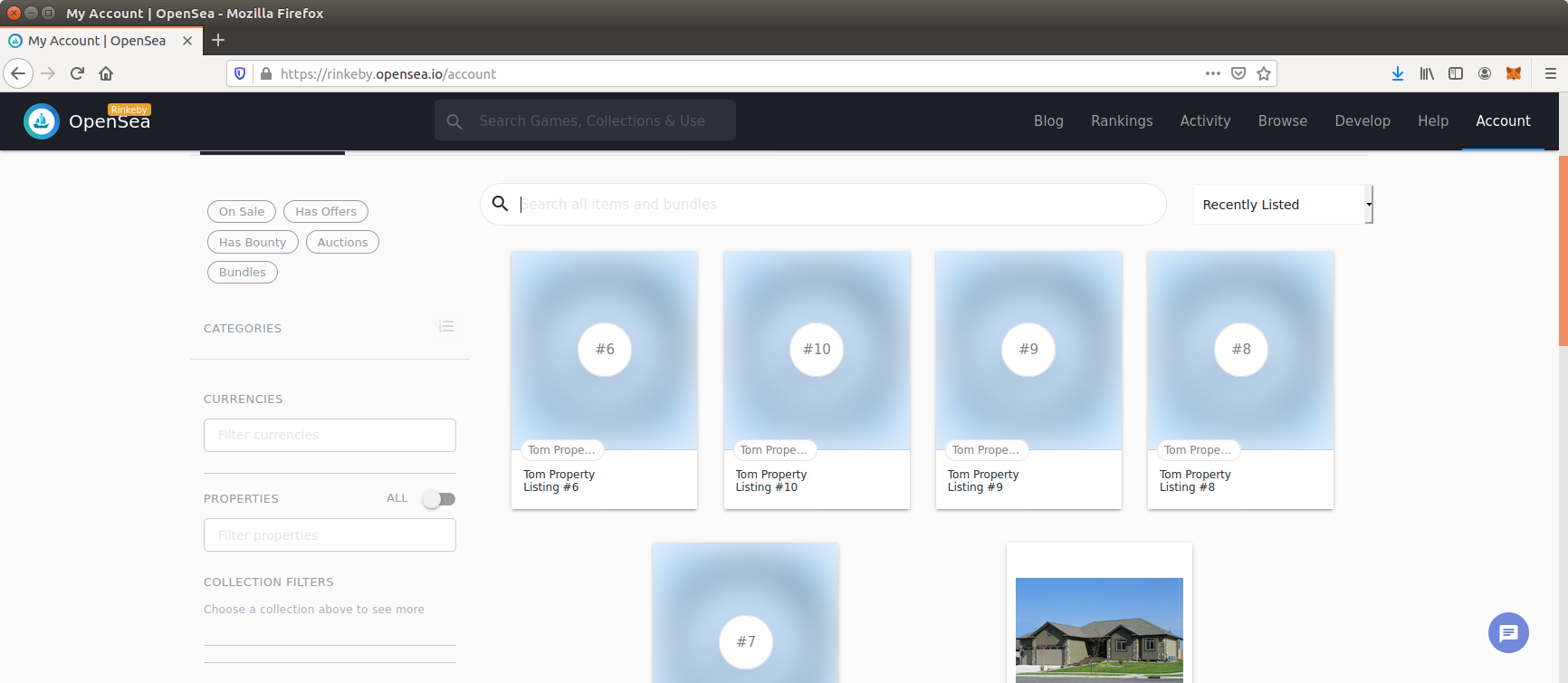


Here are the 10 tokens from Etherscan

<https://rinkeby.etherscan.io/address/0x47FB7b2E4Dfb78A90306B2feCBF16BaF0c78d7Ea#tokentxnsErc721>

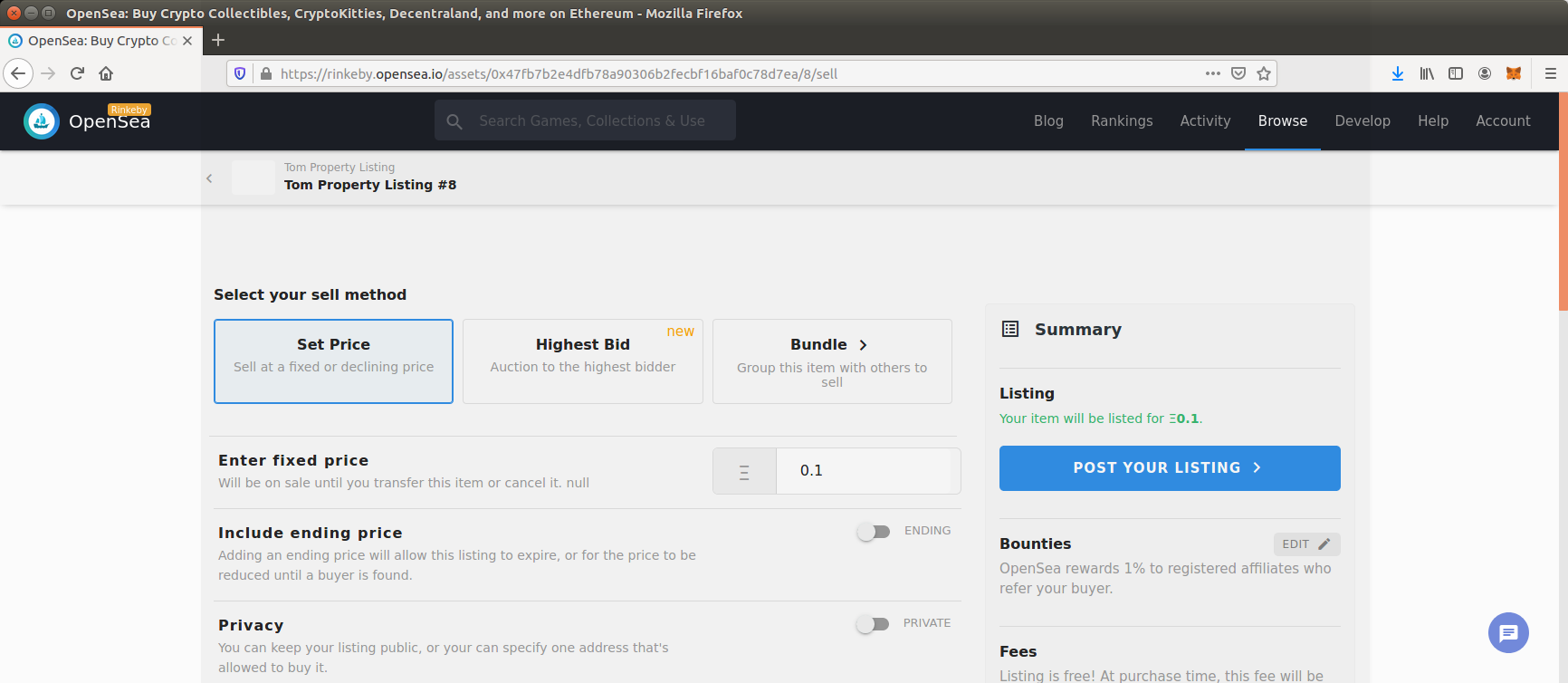


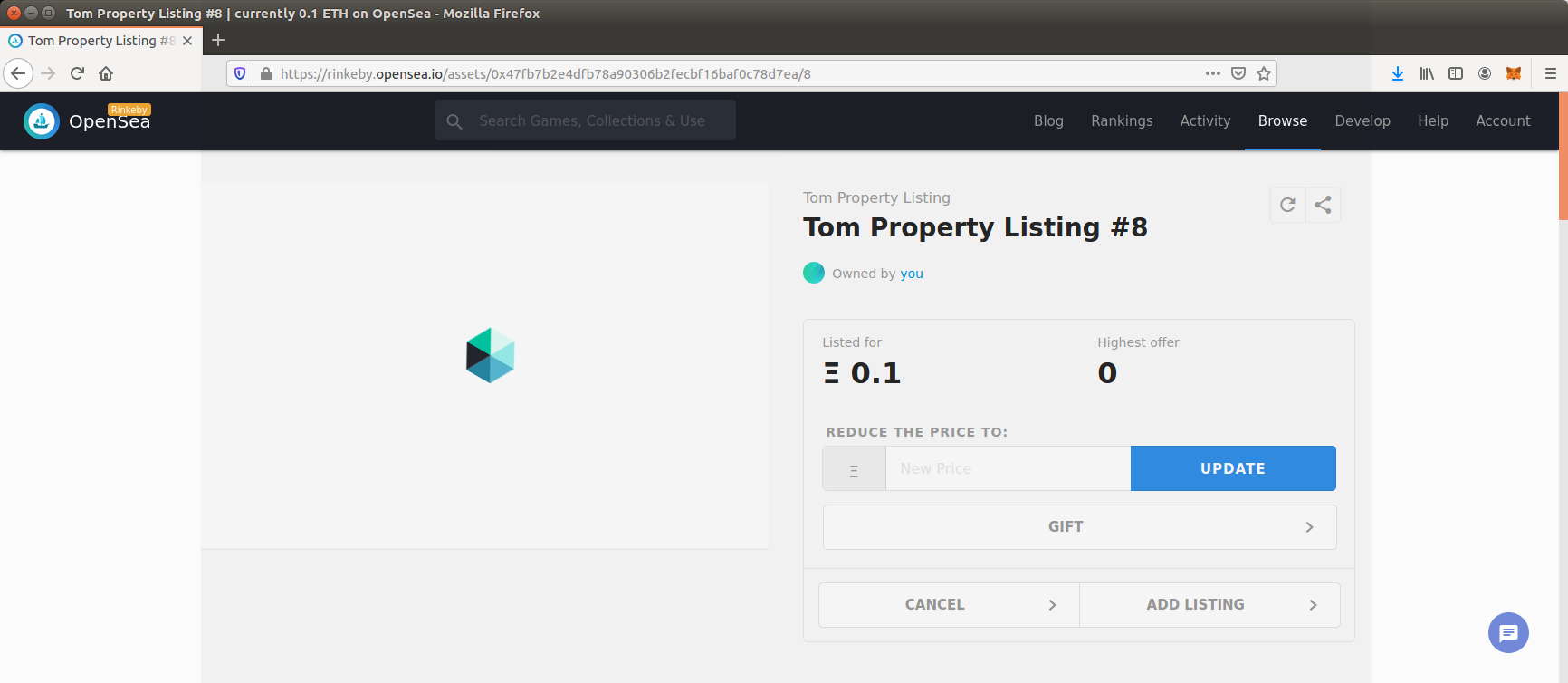
OpenSea list the TPL token 0x7734bF52F5F4C2278d3bA2B6f0C2Fa76d2356273 owns.



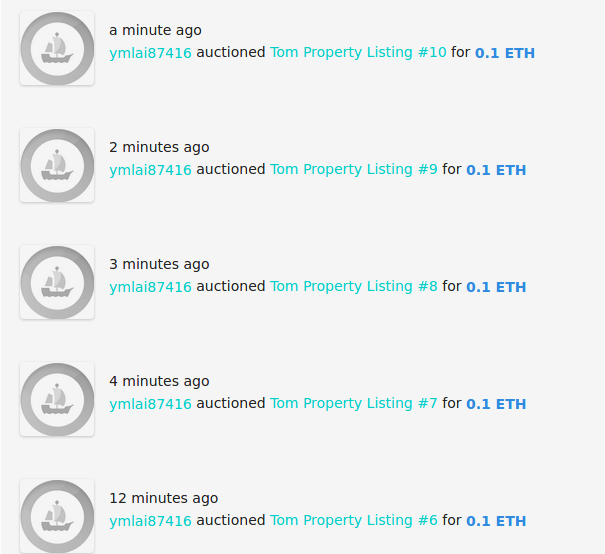
The following demo how token 6, 7, 8, 9 and 10 are sold on OpenSea Marketplace for 0.1 ETH.

Set the price for token 8 = 0.1ETH

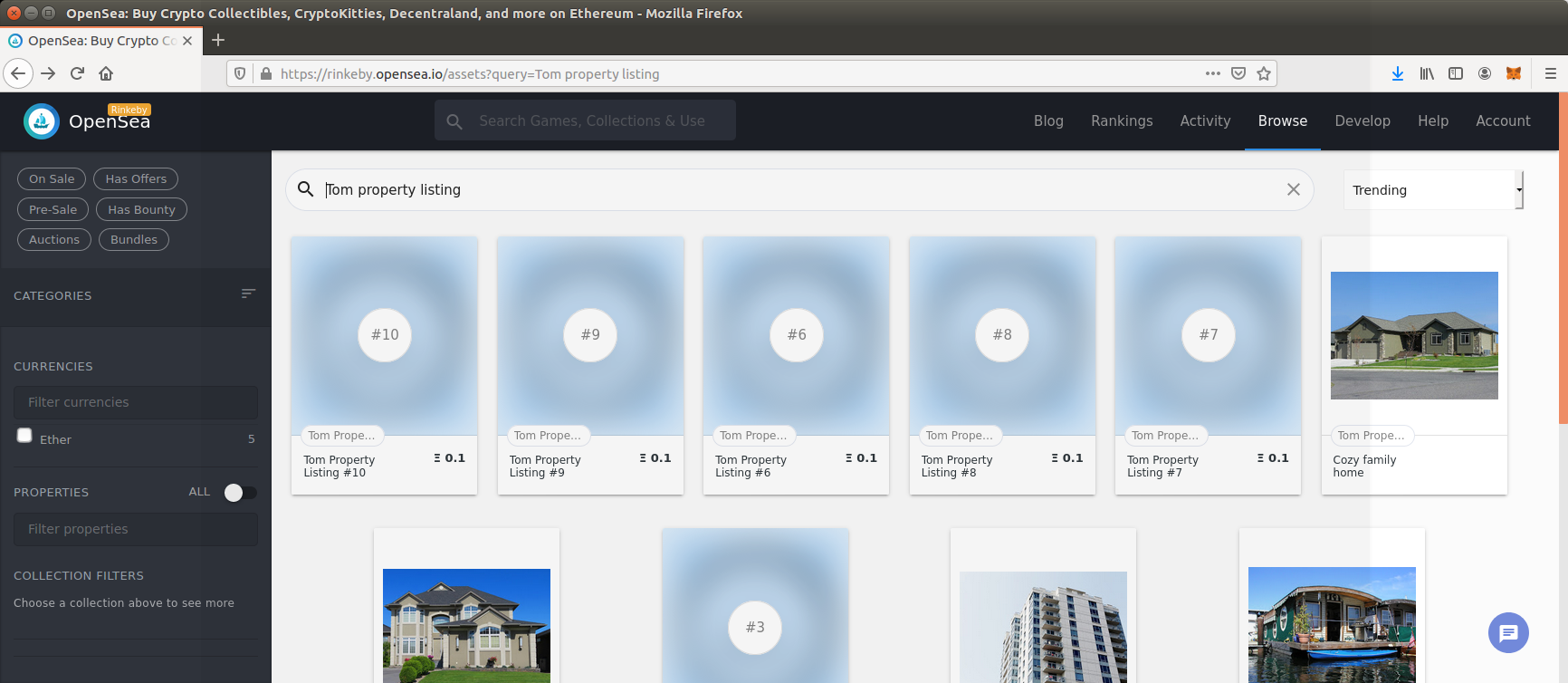




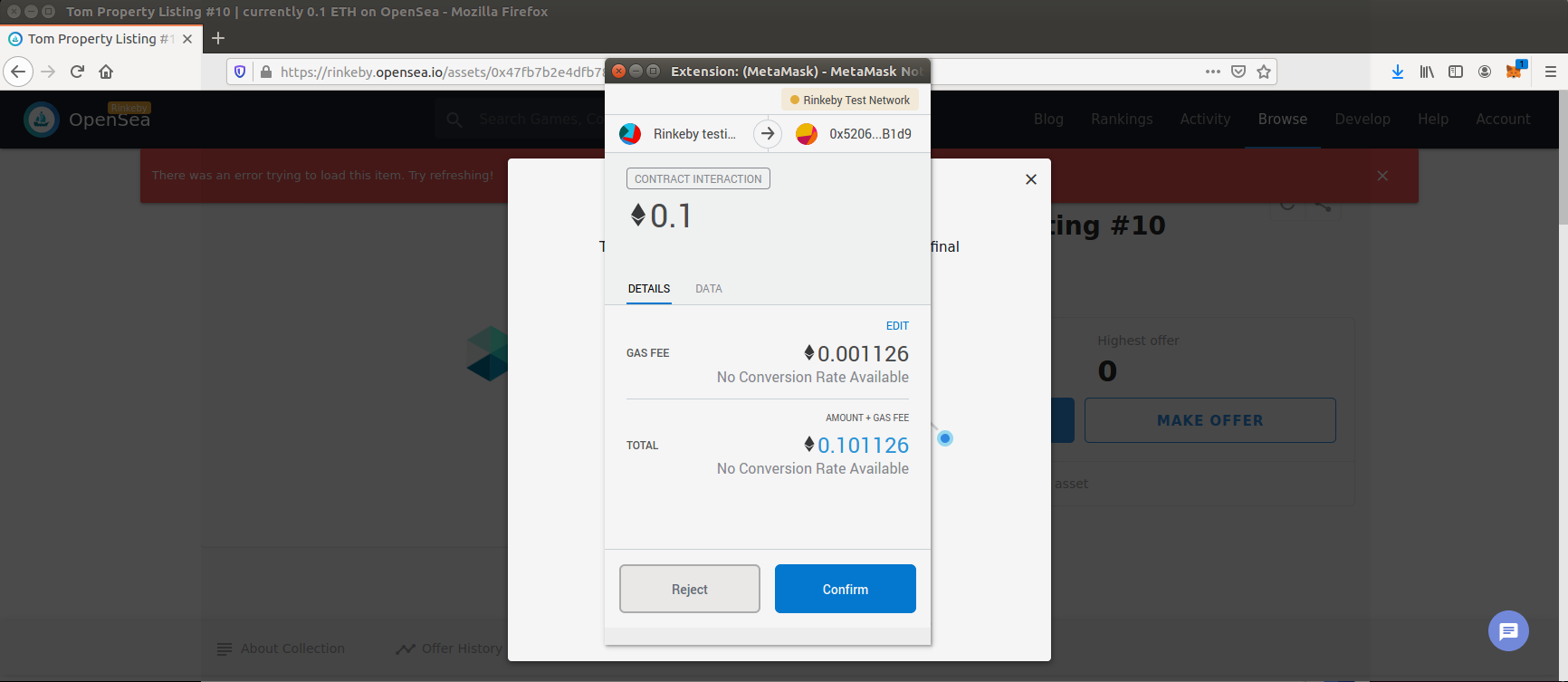
OpenSea shows that TPL token 6-10 are listed on the market.



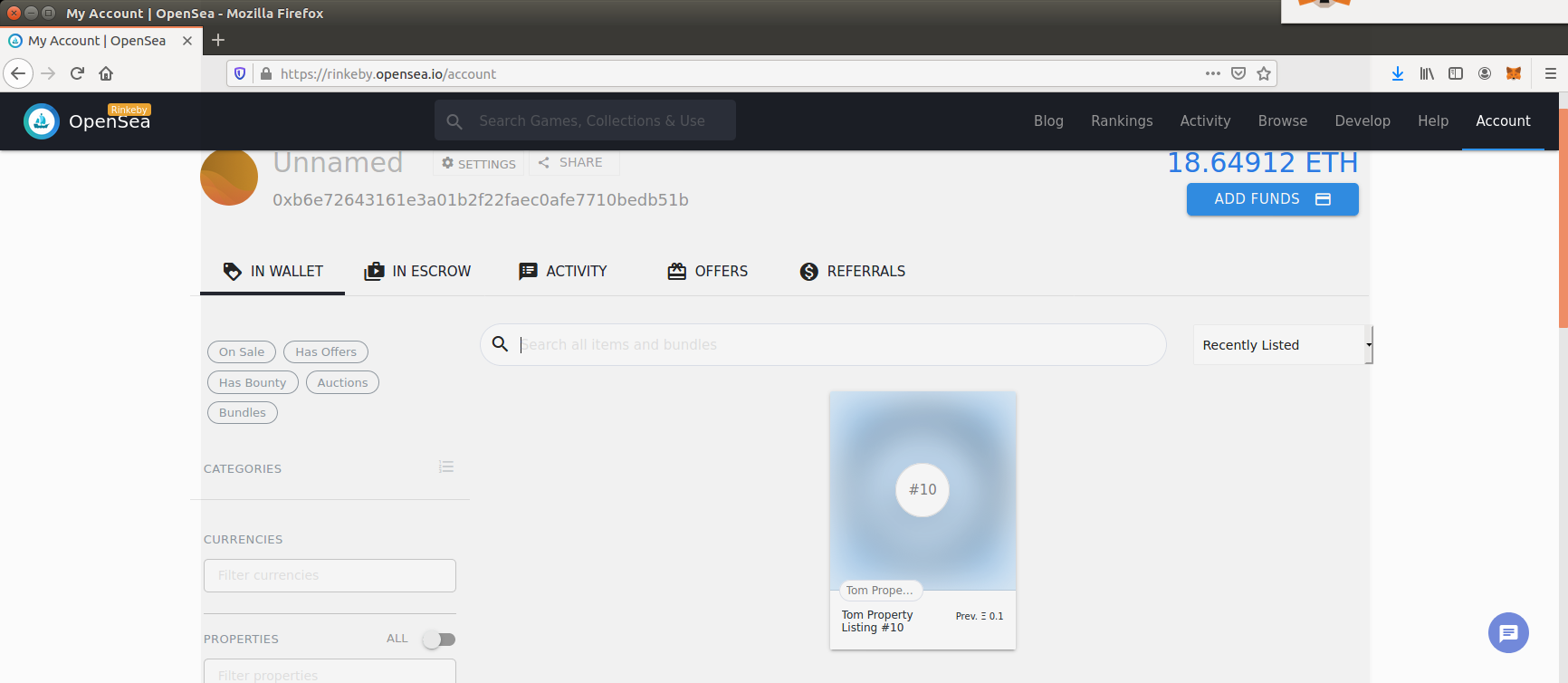
TPL token 6-10 on the market



Now another address 0xb6e72643161e3a01b2f22faec0afe7710bedb51b wants to purchase TPL token 10.



Transaction successful



Now 0xb6e72643161e3a01b2f22faec0afe7710bedb51b have all 5 tokens sold by 0x7734bf52f5f4c2278d3ba2b6f0c2fa76d2356273.

