

How to Implement an ERC721 Market

Put tokenized items for sale in a decentralized classifieds board

[Alberto Cuesta Cañada](#)

[Follow](#)

Coinmonks

--

Listen

Share

In this article, I'm going to show you how to code Craigslist for the [Ethereum blockchain](#).

Before Gumtree, eBay, and Craigslist, classified boards were mostly made of cork or paper. There were classifieds boards in school corridors, newspapers, streetlights, storefronts.

All that changed with the internet. The number of people that could see a specific classified board multiplied by many orders of magnitude. With that, the markets they represent became much more efficient and scaled to global size. eBay is a massive business that traces its origins to these physical classifieds boards.

With [blockchain](#) these markets are set to change once more, let me show you how.

Monetization

The business model of a [public blockchain](#) classifieds board will need to be different from that of eBay and the company.

First, there is the decentralization angle. Existing platforms need to maintain their own servers. A decentralized platform is maintained by its users, so the cost of running the core platform drops to zero for the platform owner.

Then there is the front end, the website or interface that gives access to the platform. Here there are many options. The platform owners can restrict access and force everyone to use their interface, charging a fee. The platform owners can also decide to open access (Power to the People!) and let anyone build interfaces to the platform. Or the owners could decide any approach in the middle of those extremes.

Business leaders with more vision than I will know how to monetize this. All I see is that this is different from the status quo and probably profitable.

Furthermore, there is an automation and payment angle. Some things can be very [effectively tokenized](#) and traded in a classifieds board. Tokenized assets are easily transferred in a blockchain. Highly complex payment methods can be easily implemented in a blockchain.

I'm just smelling a business opportunity here. A classifieds board with no running costs can easily be implemented, with complex payment paths included in each transaction. I'm sure someone will come up with an idea about what to use this for.

I'm just happy to build it. Let's have a look at the code.

Implementation

Some time ago we started an [open-source repository](#) with business case example implementations and other goodies, please have a look.

The code for this [Ethereum Classifieds Board](#) is there, please use it and abuse it. Just be aware that the code hasn't been audited and you need to do your own due diligence before letting money go into it.

The basics of the board are not complex. All the adverts on the board will be just a struct with a few fields:

So there is someone posting the advert. An item for sale. A price for the item. The status of the trade which can be open, executed or canceled.

All these trades will be kept in a mapping. Because everything in Solidity seems to be a mapping. Also because it is convenient.

Using a mapping just means that we have to come up with an id for each advert before posting it, and we will need to know

the id of an advert before we can operate on it. There are multiple ways of dealing with this either in the smart contract or in the front-end. Please ask if you need some pointers.

Next comes the question of what are those items we deal with, and what is this currency that is used to pay for the transaction.

For the items, we are just going to ask that they implement the [ERC721](#) interface, which really is just a way of representing real-world items in a blockchain, although it [works best with digital assets](#). We are going to specify our own ERC721 contract in the constructor, meaning that any assets in our classifieds board need to have been tokenized beforehand.

For the payments, we are going to do something similar. Most blockchain projects define their own [ERC20 cryptocurrency](#). Some others prefer to use a mainstream one like [DAI](#). In this classifieds board, you just need to decide on construction what your currency will be. Easy.

We are getting there. We've got adverts, items for trade and currency for payments. To make an advert means to put an item in escrow to show both that you have it and that you haven't posted it twice, possibly on a different board.

The code below does exactly that. Puts the item in escrow, creates the advert, does some housekeeping.

To accept the trade means to choose an advert (trade), pay the price, receive the item. The code below retrieves a trade. Checks it's available. Pays the item. Retrieves the item. Updates the advert.

Finally, we have an option for sellers to back out of a trade before a buyer accepts it. In some models, adverts would instead be live for a period of time before they expire. Your choice, depending on the design of your market.

The code is very similar to that used to execute a trade, only that there is no currency changing hands and the item goes back to the advert poster.

That's it. You made it to the end of the implementation. It is quite surprising how compact some business concepts are when expressed in code, and this is one of those cases. Check the complete contract [in our repo](#).

Conclusion

Classifieds boards are a common market configuration that scaled massively with the internet, becoming a hugely popular business model with a few monopolistic winners.

Classifieds boards also happen to be an easy tool to replicate in a blockchain environment, with very specific features that will make a challenge to the existing giants possible.

In this article, I made an attempt to bridge the business reality of a classifieds board business with the technological implementation. This knowledge should help you to create a vision and a roadmap for implementation if you have the right skills.

As always, if you are out to build anything fun and would welcome some advice, [please drop me a line](#)! I'm always happy to help.

[Get Best Software Deals Directly In Your Inbox](#)