Tutorial: Order Matching Bot

Introduction

Order Matching Bots (Matching Bots) are responsible for matching two orders that cross or a taker order against the AMM. Specifically, this includes:

- Market Orders
- : Market Buy and Market Sell
- Limit Orders
- : Limit Buy and Limit Sell

Matching Bots receive a small compensation for each order that they successfully fill.

SeeKeepers & Decentralised Orderbook for a technical explanation of how the decentralised orderbook (DLOB) and matching incentives work.

Matching Bots are similar toTutorial: Order Trigger Bot in that they:

- also maintain a local copy of the Decentralised Limit Orderbook (DLOB);
- do not require the operator to manage collateral; and
- receive a small reward for performing their duties.

Getting Started

The reference implementation of the Order Matching Bot is availablehere(opens in a new tab) .

Follow the instructions atKeeper Bots to set the required environment variables and make sure aClearingHouseUser is initialized.

Start the Matching Bot:

yarn run

dev:filler

Technical Explanation

1. Get nodes from the DLOB that are ready to be filled

Market orders that are sent on the Drift Protocol first go through theJust-In-Time (JIT) Auctions . After the auction period, Matching Bots step in to fill orders for a small reward.

The DLOB implementation includes a method for getting orders ready to be filled:

const

market

=

this . clearingHouse .getMarketAccounts ()[0]; // get a MarketAccount

const

oraclePriceData

=

this . driftClient .getOracleDataForMarket (marketIndex); const

oracleIsValid

=

isOracleValid (market .amm , oraclePriceData , this . driftClient .getStateAccount ().oracleGuardRails , this . slotSubscriber .getSlot ());

const

vAsk

=

calculateAskPrice (market , oraclePriceData); const

vBid

=

calculateBidPrice (market , oraclePriceData);

const

nodesToFill

=

this . dlob .findNodesToFill (marketIndex , vBid , vAsk , this . slotSubscriber .getSlot () , oracleIsValid ? oraclePriceData :

undefined);

2. Filter for Fillable Nodes

To avoid trying to fill orders that aren't ready to be filled, filter out orders that are too small to fill

if (! nodeToFill .makerNode && (isVariant (nodeToFill . node . order .orderType ,

"limit") || isVariant (nodeToFill . node . order .orderType ,

"triggerLimit"))) { const

baseAssetAmountMarketCanExecute

= calculateBaseAssetAmountMarketCanExecute (market , nodeToFill . node .order , oraclePriceData);

if (baseAssetAmountMarketCanExecute .lt (market . amm .baseAssetAmountStepSize)) { // skip order continue ; } }

3. Callfill_order

onDriftClient

const

user

=

this . userMap .get (nodeToFill . node . userAccount .toString ()); const

txSig

=

await

this . driftClient .fillOrder (nodeToFill . node .userAccount , user .getUserAccount () , nodeToFill . node .order , undefined);

Keeper Bots Tutorial: Order Trigger Bot