JIT Maker Bot

This tutorial shows how to run a JIT Maker bot in Typescript from the keeper-bots-v2 repo(opens in a new tab) . A similar example written in Python is available in keepyrs(opens in a new tab) .

Market orders go through the Just-In-Time (JIT) Auctions where Makers fight to fill orders before the order is allowed to fill against the Drift AMM .

Running the bot

☠ This bot requires collateral to run. This tutorial is a developer's guide and holds no responsibility over bot outcomes.

1. Clone the repository

The JIT Maker example lives in the keeper-bots-v2 repo, clone it to your local machine:

```
git clone https://github.com/drift-labs/keeper-bots-v2.git
```

2. Prepare a keypair and drift account.

Refer to the bot wallet setup tutorial for how to set up a bot wallet.

3. Prepare the config file.

The jit-maker-config.yaml (opens in a new tab) file is good starting point. You will need to fill in the following values:

- global.endpoint
- : set this to your RPC endpoint (see RPC Providers
-)
- global.keeperPrivateKey
- : set this to your bot private key (alternatively you can leave it blank and set the KEEPER_PRIVATE_KEY
- environment variable)
- botConfigs.jitMaker.aggressivenessBps
- : this is how aggressive you want your jit maker to be. If set to 30, the bot will attempt to buy 30 bps above the best bid, and sell 30 bps below the best ask.
- botConfigs.jitMaker.marketType
- (perp
- orspot
-): this is the type of market we want to jit make
- botConfigs.jitMaker.marketIndexes
- : this is the list of markets that we want to jit make, it
- botConfigs.jitMaker.subaccounts
- : this is the subaccount to use for each marketIndex provided

Note:subaccounts andmarketIndexes are a direct mapping, so the below config will use subaccount 0 for marketIndex 0, and subaccount 1 for marketIndex 1:

```
botConfigs : jitMaker : marketType :

perp marketIndexes : - 0 - 1 subaccounts : - 0 - 1
```

4. Run the bot

Now you are ready to run the bot, start it with:

```
yarn run dev --config-file=jit-maker-config.yaml
```

Technical Explanation

Strategy overview

This bot uses the JitterShotgun (opens in a new tab) strategy and the jit-proxy (opens in a new tab) program to try to fill jit maker fills.

The JitterShotgun strategy continuously sends transactions in an attempt to fill orders as soon as it sees one that crosses. Thejit-proxy program is a permissionless and stateless program that does the last-mile checks on-chain to ensure the fill is within our desired bid/ask price and does not exeeed min/max positions.

JIT-able orders

Orders withauctionDuration > 0 may be filled by jit makers at any time. This bot finds these orders by using theprogramSubscribe RPC method, and filtering for users with new orders that meet this criteria.

Final notes and future optimizations

This is a very simple and naiive strategy, some oppotunities for improvements:

- allow multiple markets and spot + perp at the same time
- asymetric bid/ask offsets, allow the differentaggressivenessBps
- for bid/ask depending on how you want to skew your quotes
- an improvedJitterShotgun
- strategy that is more cognizant of time and transactions sent

Troubleshooting

Resubscribing log messages

No ws data from user in 30000ms, resubscribing No ws data from userStats in 30000ms, resubscribing No ws data from perpMarket in 30000ms, resubscribing No ws data from perpMarket in 30000ms, resubscribing No ws data from perpMarket in 30000ms, resubscribing No ws data from spotMarket in 30000ms, resubscribing This is a notification from the Drift SDK that it is restarting its websocket connection with the RPC due to no messages being received within the set time. This is generatlly not an error and pretty common for less active markets that don't have much activity.

Running JIT periodic tasks...

[2024-02-27T00:04:31.387Z] Running JIT periodic tasks... [2024-02-27T00:04:31.389Z] info: (mkt index: JTO-PERP) base to market make (targetLvg=0.95): 1481.8891930588115 = 3476.551044 / 2.228725 * 0.95 This is a normal status message that the bot is running its periodic tasks. If you see this message, it means the bot is running as expected.

Order does not cross params yet, retrying

Trying to fill 24yfkkuFv849BpCBBo49tV6i6ycfc3aEUY6wWKFsN4SS-177961 SendTransactionError: failed to send transaction: Transaction simulation failed: Error processing Instruction 2: custom program error: 0x1771 ... logs: ['Program ComputeBudget111111111111111111111111111111 invoke [1]', 'Program ComputeBudget111111111111111111111111111111 success', 'Program ComputeBudget111111111111111111111111111111 invoke [1]', 'Program ComputeBudget111111111111111111111111111111 success', 'Program J1TnP8zvVxbtF5KFp5xRmWuvG9McnhzmBd9XGfCyuxFP invoke [1]', 'Program log: Instruction: Jit', 'Program log: slot = 250667476 auction duration = 32 slots_left = 15', 'Program log: taker order type Oracle auction start -12100 auction end -200 limit price 0 oracle price offset -193', 'Program log: taker price 2220700 < worst ask 2234257', 'Program log: AnchorError occurred. Error Code: AskNotCrossed. Error Number: 6001. Error Message: AskNotCrossed.', 'Program J1TnP8zvVxbtF5KFp5xRmWuvG9McnhzmBd9XGfCyuxFP consumed 39672 of 599700 compute units', 'Program J1TnP8zvVxbtF5KFp5xRmWuvG9McnhzmBd9XGfCyuxFP failed: custom program error: 0x1771']

Failed to fill 24yfkkuFv849BpCBBo49tV6i6ycfc3aEUY6wWKFsN4SS-177961 Order does not cross params yet, retrying This is a failed tx message from the RPC provider when a transaction is sent. It is common for the shotgun strategy to get this message as it sends many transactions at once.

Decoding it further:

- It is a jit fill attempt on taker24yfkkuFv849BpCBBo49tV6i6ycfc3aEUY6wWKFsN4SS
- 's orderId:177961
- the current slot (that the RPC is simulating the transaction in) is 250667476, the auction lasts for 32 slots, and there are 15 slots left
- the taker's order is an Oracle order, with an offset of -12100 (-0.0121) to -200 (-0.0020)
- the taker's order is abuy
- (since it is comparing with ourask
- price)
- the taker's price at that slot is 2220700 (2.2207), which is below our worst ask 2234257 (2.2343), so the jit-proxy program threw and error to prevent us from sending a failing transaction on chain

Tutorial: Order Liquidation Bot RPC Providers