Isolated Markets

In v5.0.0 the Isolated Markets feature was added to the V4 chain software. The below is an overview of how trading will work on Isolated Markets on the V4 chain software.

Note: This document covers how the feature works from the protocol point of view and not the front-end or the indexer. For more details on what isolated markets are see theblog post(opens in a new tab).

Trading Isolated Markets

Positions in isolated markets can only be opened on a subaccount with no open perpetual positions or a subaccount with an existing perpetual position in the same isolated market. Once a perpetual position for an isolated market is opened on a subaccount, no positions in any other markets can be opened until the perpetual position is closed.

The above restriction only applies to positions, orders can still be placed for different markets on a subaccount while it holds an open position for an isolated market. The orders will fail and be canceled when they match if the subaccount still holds an open position for a different isolated market. A newerror code(opens in a new tab) 2005 has been added to indicate such a failure.

Other than the above caveat, isolated markets can be traded in the same way as before v5.0.0.

Note: The maximum number of subaccounts per address was increased from 127 to 128000 in v5.0.0 to address the need for a separate subaccount per isolated market.

Querying for Isolated Markets

There is a newmarket_type parameter in thePerpetualParams proto struct that indicates the type of market.

There are 2 possible values for this parameter:

- PERPETUAL_MARKET_TYPE_CROSS
-
    - markets where subaccounts can have positions cross-margined with otherPERPETUAL_MARKET_TYPE_CROSS
- markets, all markets created before the v5.0.0 upgrade arePERPETUAL_MARKET_TYPE_CROSS
- markets
- PERPETUAL_MARKET_TYPE_ISOLATED
-
    - markets that can only be margined in isolated, no cross-margining with other markets is possible

An example of how each type of market looks when queried using the/dydxprotocol/perpetuals/perpetual/:id REST endpoint.

- PERPETUAL_MARKET_TYPE_CROSS

{ "perpetual" : { "params" : { "id" :

1 , "ticker" :

"ETH-USD" , "market_id" :

1 , "atomic_resolution" :

-9 , "default_funding_ppm" :

0 , "liquidity_tier" :

0 , "market_type" :

"PERPETUAL_MARKET_TYPE_CROSS" } , "funding_index" :

"0" , "open_interest" :

"0" } } * PERPETUAL_MARKET_TYPE_ISOLATED

{ "perpetual" : { "params" : { "id" :

1 , "ticker" :

"FLOKI-USD" , "market_id" :

37 , "atomic_resolution" :

-13 , "default_funding_ppm" :

0 , "liquidity_tier" :

2 , "market_type" :

"PERPETUAL_MARKET_TYPE_ISOLATED" } , "funding_index" :

"0" , "open_interest" :

"0" } } Last updated onJune 6, 2024Short-term vs Stateful CLI Commands