

Client

Queries

Params

Queries the parameters of the module.

GET "/dex/params" Proto Messages

Request:

message

QueryParamsRequest

{ } Response:

message

QueryParamsResponse

{

// params holds all the parameters of this module. Params params =

1

[(gogoproto . nullable)

=

false] ; } Curl Command (testnet):

curl https://rest-falcron.pion-1.ntrn.tech/neutron/dex/params

LimitOrderTrancheUser

GET "/dex/limit_order_tranche_user/{address}/{tranche_key}" This query retrieves aLimitOrderTrancheUser by user address and TrancheKey.

Proto Messages

Request:

message

QueryGetLimitOrderTrancheUserRequest

{ string address =

1 ; string tranche_key =

2 ; } Response:

message

QueryGetLimitOrderTrancheUserResponse

{ LimitOrderTrancheUser limit_order_tranche_user =

1

[(gogoproto . nullable)

=

true] ; }

message

```

LimitOrderTrancheUser
{ TradePairID trade_pair_id =
1 ; int64 tick_index_taker_to_maker =
2 ; string tranche_key =
3 ; string address =
4 ; string shares_owned =
5
[ ( gogoproto . moretags )
=
"yaml:\\"shares_owned\\"\" , ( gogoproto . customtype )
=
"github.com/cosmos/cosmos-sdk/types.Int" , ( gogoproto . nullable )
=
false , ( gogoproto . jsontag )
=
"shares_owned" ] ; string shares_withdrawn =
6
[ ( gogoproto . moretags )
=
"yaml:\\"shares_withdrawn\\"\" , ( gogoproto . customtype )
=
"github.com/cosmos/cosmos-sdk/types.Int" , ( gogoproto . nullable )
=
false , ( gogoproto . jsontag )
=
"shares_withdrawn" ] ; string shares_cancelled =
7
[ ( gogoproto . moretags )
=
"yaml:\\"shares_cancelled\\"\" , ( gogoproto . customtype )
=
"github.com/cosmos/cosmos-sdk/types.Int" , ( gogoproto . nullable )
=
false , ( gogoproto . jsontag )
=
"shares_cancelled" ] ; LimitOrderType order_type =
8 ; } Arguments

```

- QueryGetLimitOrderTrancheUserRequest
- : Request message for theLimitOrderTrancheUser
- query.* address
- - (string): The user address.
- - tranche_key
- - (string): The tranche key.

Sample Query

Curl Command (testnet):

curl https://rest-falcon.pion-1.ntrn.tech/neutron/dex/limit_order_tranche_user/ { address } / { tranche_key }

LimitOrderTrancheUserAll

GET "/dex/Neutron DEX/dex/limit_order_tranche_user" This query retrieves a list ofLimitOrderTrancheUser items.

Proto Messages

Request:

message

QueryAllLimitOrderTrancheUserRequest

{ cosmos . base . query . v1beta1 . PageRequest pagination =
1 ; } Response:

message

QueryAllLimitOrderTrancheUserResponse

{ repeated

LimitOrderTrancheUser limit_order_tranche_user =

1

[(gogoproto . nullable)

=

true] ; cosmos . base . query . v1beta1 . PageResponse pagination =

2 ; } Arguments

- QueryAllLimitOrderTrancheUserRequest
- : Request message for theLimitOrderTrancheUserAll
- query.* pagination
- - (cosmos.base.query.v1beta1.PageRequest): Pagination options.

Sample Query

Curl Command (testnet):

curl https://rest-falcon.pion-1.ntrn.tech/dex/limit_order_tranche_user

LimitOrderTranche

GET "/dex/limit_order_tranche/{pairID}/{tokenIn}/{tickIndex}/{trancheKey}" This query retrieves aLimitOrderTranche by a tranche's key (pairID + tokenIn + tickIndex + trancheKey).

Proto Messages

Request:

message

QueryGetLimitOrderTrancheRequest

```
{ string pair_id =  
1 ; int64 tick_index =  
2 ; string token_in =  
3 ; string tranche_key =  
4 ; } Response:
```

message

QueryGetLimitOrderTrancheResponse

```
{ LimitOrderTranche limit_order_tranche =  
1  
[ ( gogoproto . nullable )  
=  
true ] ; }
```

message

LimitOrderTrancheKey

```
{ TradePairID trade_pair_id =  
1 ; int64 tick_index_taker_to_maker =  
2 ; string tranche_key =  
3 ; }
```

message

LimitOrderTranche

```
{ LimitOrderTrancheKey key =  
1 ; string reserves_maker_denom =  
2  
[ ( gogoproto . moretags )  
=  
"yaml:\\"reserves_maker_denom\\"\" , ( gogoproto . customtype )  
=  
"github.com/cosmos/cosmos-sdk/types.Int" , ( gogoproto . nullable )  
=  
false , ( gogoproto . jsontag )  
=  
"reserves_maker_denom" ] ; string reserves_taker_denom =  
3  
[ ( gogoproto . moretags )  
=  
"yaml:\\"reserves_taker_denom\\"\" , ( gogoproto . customtype )
```

```

=
"github.com/cosmos/cosmos-sdk/types.Int" , ( gogoproto . nullable )
=
false , ( gogoproto . jsontag )
=
"reserves_taker_denom" ] ; string total_maker_denom =
4
[ ( gogoproto . moretags )
=
"yaml:\"total_maker_denom\"" , ( gogoproto . customtype )
=
"github.com/cosmos/cosmos-sdk/types.Int" , ( gogoproto . nullable )
=
false , ( gogoproto . jsontag )
=
"total_maker_denom" ] ; string total_taker_denom =
5
[ ( gogoproto . moretags )
=
"yaml:\"total_taker_denom\"" , ( gogoproto . customtype )
=
"github.com/cosmos/cosmos-sdk/types.Int" , ( gogoproto . nullable )
=
false , ( gogoproto . jsontag )
=
"total_taker_denom" ] ; // expiration_time is represented as an RFC 3339 formatted date. // LimitOrders with expiration_time
set are valid as long as blockTime <= expiration_time. // JIT orders also use expiration_time to handle deletion, but
represent a special case. // All JIT orders have an expiration_time of 0001-01-01T00:00:00Z, and an exception is made to //
still treat these orders as live. Order deletion still functions the // same, and the orders will be deleted at the end of the block.
google . protobuf . Timestamp expiration_time =
6
[ ( gogoproto . stdtime )
=
true , ( gogoproto . nullable )
=
true ] ; string price_taker_to_maker =
7
[ ( gogoproto . moretags )
=

```

```
"yaml:\\"price_taker_to_maker\\"" , ( gogoproto . customtype )
=
"github.com/neutron-org/neutron/v2/utls/math.PrecDec" , ( gogoproto . nullable )
=
false , ( gogoproto . jsontag )
=
"price_taker_to_maker" ] ; } Arguments
```

- QueryGetLimitOrderTrancheRequest
- : Request message for theLimitOrderTranche
- query.* pairID
- - (string): The pair ID.
- - tickIndex
- - (int64): The tick index.
- - tokenIn
- - (string): The input token.
- - trancheKey
- - (string): The tranche key.

Sample Query

Curl Command (testnet):

```
curl https://rest-falcron.pion-1.ntrn.tech/neutron/dex/limit_order_tranche/ { pairID } / { tokenIn } / { tickIndex } / { trancheKey }
```

LimitOrderTrancheAll

GET "dex/limit_order_tranche/{pairID}/{tokenIn}" This query retrieves a list ofLimitOrderTranche items for a given pairID / TokenIn combination.

Proto Messages

Request:

message

QueryAllLimitOrderTrancheRequest

```
{ string pair_id =
1 ; string token_in =
2 ; cosmos . base . query . v1beta1 . PageRequest pagination =
3 ; } Response:
```

message

QueryAllLimitOrderTrancheResponse

```
{ repeated
```

```
LimitOrderTranche limit_order_tranche =
```

```
1
```

```
[ ( gogoproto . nullable )
```

```
=
```

true] ; cosmos . base . query . v1beta1 . PageResponse pagination =

2 ; } Arguments

- QueryAllLimitOrderTrancheRequest
- : Request message for theLimitOrderTrancheAll
- query.* pairID
- - (string): The pair ID.
- - tokenIn
- - (string): The input token.
- - pagination
- - (cosmos.base.query.v1beta1.PageRequest): Pagination options.

Sample Query

Curl Command (testnet):

curl https://rest-falcron.pion-1.ntrn.tech/neutron/dex/limit_order_tranche/ { pairID } / { tokenIn }

UserDepositsAll

GET "/dex/user/deposits/{address}" This query retrieves a list ofDepositRecord items by user address.

Proto Messages

Request:

message

QueryAllUserDepositsRequest

{ string address =

1 ; cosmos . base . query . v1beta1 . PageRequest pagination =

2 ; } Response:

message

QueryAllUserDepositsResponse

{ repeated

DepositRecord deposits =

1

[(gogoproto . nullable)

=

true] ; cosmos . base . query . v1beta1 . PageResponse pagination =

2 ; } Arguments

- QueryAllUserDepositsRequest
- : Request message for theUserDepositsAll
- query.* address
- - (string): The user address.
- - pagination
- - (cosmos.base.query.v1beta1.PageRequest): Pagination options.

Sample Query

Curl Command (testnet):

```
curl https://rest-falcon.pion-1.ntrn.tech/neutron/dex/user/deposits/ { address }
```

LimitOrderTrancheUserAllByAddress

GET "/neutron/dex/user/limit_orders/{address}" This query retrieves a list of LimitOrderTrancheUser items by user address.

Proto Messages

Request:

message

QueryAllUserLimitOrdersRequest

{ string address =

1 ; cosmos . base . query . v1beta1 . PageRequest pagination =

2 ; } Response:

message

QueryAllUserLimitOrdersResponse

{ repeated

LimitOrderTrancheUser limit_orders =

1

[(gogoproto . nullable)

=

true] ; cosmos . base . query . v1beta1 . PageResponse pagination =

2 ; } Arguments

- QueryAllUserLimitOrdersRequest
- : Request message for the UserLimitOrdersAll
- query.* address
- - (string): The user address.
- - pagination
- - (cosmos.base.query.v1beta1.PageRequest): Pagination options.

Sample Query

Curl Command (testnet):

```
curl https://rest-falcon.pion-1.ntrn.tech/neutron/dex/user/limit_orders/ { address }
```

TickLiquidityAll

GET "/neutron/dex/tick_liquidity/{pairID}/{tokenIn}" This query retrieves a list of TickLiquidity items for a given pairID / TokenIn combination.

Proto Messages

Request:

message

QueryAllTickLiquidityRequest

{ string pair_id =

1 ; string token_in =


```

2 ; cosmos . base . query . v1beta1 . PageRequest pagination =
3 ; } Response:
message
TickLiquidity
{ oneof liquidity { PoolReserves pool_reserves =
1 ; LimitOrderTranche limit_order_tranche =
2 ; } }
message
QueryAllTickLiquidityResponse
{ repeated
TickLiquidity tick_liquidity =
1
[ ( gogoproto . nullable )
=
true ] ; cosmos . base . query . v1beta1 . PageResponse pagination =
2 ; } Arguments

```

- QueryAllTickLiquidityRequest
- : Request message for theTickLiquidityAll
- query.* pairID
- - (string): The pair ID.
- - tokenIn
- - (string): The input token.
- - pagination
- - (cosmos.base.query.v1beta1.PageRequest): Pagination options.

Sample Query

Curl Command (testnet):

```
curl https://rest-falcron.pion-1.ntrn.tech/neutron/dex/tick_liquidity/ { pairID } / { tokenIn }
```

InactiveLimitOrderTranche

GET "/neutron/dex/filled_limit_order_tranche/{pairID}/{tokenIn}/{tickIndex}/{trancheKey}" This query retrieves an inactiveLimitOrderTranche by a tranche's key (pairID + tokenIn + tickIndex + trancheKey).

Proto Messages

Request:

message

QueryGetInactiveLimitOrderTrancheRequest

```

{ string pair_id =
1 ; string token_in =
2 ; int64 tick_index =
3 ; string tranche_key =

```

4 ; } Response:

message

QueryGetInactiveLimitOrderTrancheResponse

{ LimitOrderTranche inactive_limit_order_tranche =

1

[(gogoproto . nullable)

=

true] ; } Arguments

- QueryGetInactiveLimitOrderTrancheRequest
- : Request message for theInactiveLimitOrderTranche
- query.* pairID
- - (string): The pair ID.
- - tokenIn
- - (string): The input token.
- - tickIndex
- - (int64): The tick index.
- - trancheKey
- - (string): The tranche key.

Sample Query

Curl Command (testnet):

```
curl https://rest-falcron.pion-1.ntrn.tech/neutron/dex/filled_limit_order_tranche/ { pairID } / { tokenIn } / { tickIndex } / { trancheKey }
```

InactiveLimitOrderTrancheAll

GET "/neutron/dex/filled_limit_order_tranche" This query retrieves a list of inactiveLimitOrderTranche items.

Proto Messages

Request:

message

QueryAllInactiveLimitOrderTrancheRequest

{ cosmos . base . query . v1beta1 . PageRequest pagination =

1 ; } Response:

message

QueryAllInactiveLimitOrderTrancheResponse

{ repeated

LimitOrderTranche inactive_limit_order_tranche =

1

[(gogoproto . nullable)

=

true] ; cosmos . base . query . v1beta1 . PageResponse pagination =

2 ; } Arguments

- QueryAllInactiveLimitOrderTrancheRequest
- : Request message for theInactiveLimitOrderTrancheAll
- query.* pagination
- - (cosmos.base.query.v1beta1.PageRequest): Pagination options.

Sample Query

Curl Command (testnet):

curl https://rest-falcon.pion-1.ntrn.tech/neutron/dex/filled_limit_order_tranche

PoolReservesAll

GET "/neutron/dex/pool_reserves/{pairID}/{tokenIn}" This query retrieves a list ofPoolReserves items for a given pairID / TokenIn combination.

Proto Messages

Request:

message

QueryAllPoolReservesRequest

{ string pair_id =

1 ; string token_in =

2 ; cosmos . base . query . v1beta1 . PageRequest pagination =

3 ; } Response:

message

QueryAllPoolReservesResponse

{ repeated

PoolReserves pool_reserves =

1

[(gogoproto . nullable)

=

true] ; cosmos . base . query . v1beta1 . PageResponse pagination =

2 ; } Arguments

- QueryAllPoolReservesRequest
- : Request message for thePoolReservesAll
- query.* pairID
- - (string): The pair ID.
- - tokenIn
- - (string): The input token.
- - pagination
- - (cosmos.base.query.v1beta1.PageRequest): Pagination options.

Sample Query

Curl Command (testnet):

curl https://rest-falcon.pion-1.ntrn.tech/neutron/dex/pool_reserves/ { pairID } / { tokenIn }

PoolReserves

GET "/neutron/dex/pool_reserves/{pairID}/{tokenIn}/{tickIndex}/{fee}" This query retrieves aPoolReserves by PoolReservesKey (PairID+TokenIn+TickIndex+Fee).

Proto Messages

Request:

message

QueryGetPoolReservesRequest

```
{ string pair_id =  
1 ; string token_in =  
2 ; int64 tick_index =  
3 ; uint64 fee =  
4 ; }
```

Response:

message

QueryGetPoolReservesResponse

```
{ PoolReserves pool_reserves =  
1  
[ ( gogoproto . nullable )  
=  
true ] ; }
```

Arguments

- QueryGetPoolReservesRequest
- : Request message for thePoolReserves
- query.* pairID
- - (string): The pair ID.
- - tokenIn
- - (string): The input token.
- - tickIndex
- - (int64): The tick index.
- - fee
- - (uint64): The fee.

Sample Query

Curl Command (testnet):

```
curl https://rest-falcon.pion-1.ntrn.tech/neutron/dex/pool_reserves/ { pairID } / { tokenIn } / { tickIndex } / { fee }
```

QueryEstimateMultiHopSwap

GET "/neutron/dex/estimate_multi_hop_swap" Queries the simulated result of a multihop swap

Proto Messages

Request:

message

QueryEstimateMultiHopSwapRequest

```

{ string creator =
1 ; string receiver =
2 ; repeated
MultiHopRoute routes =
3 ; string amount_in =
4
[ ( gogoproto . moretags )
=
"yaml:\\"amount_in\\"\" , ( gogoproto . customtype )
=
"github.com/cosmos/cosmos-sdk/types.Int" , ( gogoproto . nullable )
=
false , ( gogoproto . jsontag )
=
"amount_in" ] ; string exit_limit_price =
5
[ ( gogoproto . moretags )
=
"yaml:\\"exit_limit_price\\"\" , ( gogoproto . customtype )
=
"github.com/neutron-org/neutron/v2/utils/math.PrecDec" , ( gogoproto . nullable )
=
false , ( gogoproto . jsontag )
=
"exit_limit_price" ] ;

// If pickBestRoute == true then all routes are run and the route with the // best price is chosen otherwise, the first succesful
route is used. bool pick_best_route =

6 ; }

message
MultiHopRoute
{ repeated
string hops =
1 ; } Response:

message
QueryEstimateMultiHopSwapResponse
{ cosmos . base . v1beta1 . Coin coin_out =
1
[ ( gogoproto . nullable )

```

=

false , (gogoproto . customtype)

=

"github.com/cosmos/cosmos-sdk/types.Coin" , (gogoproto . jsontag)

=

"coin_out"] ; } Arguments

- QueryEstimateMultiHopSwapRequest
- : Request message for theEstimateMultiHopSwap
- query.* creator
- - (string): creator.
- - receiver
- - (string): receiver.
- - MultiHopRoute
- - ([]MultiHopRoute): Array of possible routes.
- - AmountIn
- - (sdk.Int): Amount of TokenIn to swap.
- - ExitLimitPrice
- - (sdk.Dec): Minimum price that must be satisfied for a route to succeed.
- - PickBestRoute
- - (bool): When true, all routes are run and the route with the best price is used.

Sample Query

Curl Command (testnet):

curl https://rest-falcron.pion-1.ntrn.tech/neutron/dex/estimate_multi_hop_swap

EstimatePlaceLimitOrder

GET "/neutron/dex/estimate_place_limit_order" Queries the simulated result of a limit order placement.

Proto Messages

Request:

message

QueryEstimatePlaceLimitOrderRequest

{ string creator =

1 ; string receiver =

2 ; string token_in =

3 ; string token_out =

4 ; int64 tick_index_in_to_out =

5 ; string amount_in =

6

[(gogoproto . moretags)

```

=
"yaml:\\"amount_in\\"\" , ( gogoproto . customtype )
=
"github.com/cosmos/cosmos-sdk/types.Int" , ( gogoproto . nullable )
=
false , ( gogoproto . jsontag )
=
"amount_in" ] ; LimitOrderType order_type =
7 ;
// expirationTime is only valid iff orderType == GOOD_TIL_TIME. google . protobuf . Timestamp expiration_time =
8
[ ( gogoproto . stdtime )
=
true , ( gogoproto . nullable )
=
true ] ; string maxAmount_out =
9
[ ( gogoproto . moretags )
=
"yaml:\\"max_amount_out\\"\" , ( gogoproto . customtype )
=
"github.com/cosmos/cosmos-sdk/types.Int" , ( gogoproto . nullable )
=
true , ( gogoproto . jsontag )
=
"max_amount_out" ] ; } enum
LimitOrderType { GOOD_TIL_CANCELLED =
0 ; FILL_OR_KILL =
1 ; IMMEDIATE_OR_CANCEL =
2 ; JUST_IN_TIME =
3 ; GOOD_TIL_TIME =
4 ; } Response:
message
QueryEstimatePlaceLimitOrderResponse
{ // Total amount of coin used for the limit order // You can derive makerLimitInCoin using the equation: totalInCoin = //
swapInCoin + makerLimitInCoin cosmos . base . v1beta1 . Coin total_in_coin =
1
[ ( gogoproto . moretags )

```

```

=
"yaml:\\"total_in_coin\\"", ( gogoproto . nullable )

=
false , ( gogoproto . customtype )

=
"github.com/cosmos/cosmos-sdk/types.Coin" , ( gogoproto . jsontag )

=
"total_in_coin" ] ;

// Total amount of the token in that was immediately swapped for swapOutCoin cosmos . base . v1beta1 . Coin
swap_in_coin =

2

[ ( gogoproto . moretags )

=
"yaml:\\"swap_in_coin\\"", ( gogoproto . nullable )

=
false , ( gogoproto . customtype )

=
"github.com/cosmos/cosmos-sdk/types.Coin" , ( gogoproto . jsontag )

=
"swap_in_coin" ] ;

// Total amount of coin received from the taker portion of the limit order // This is the amount of coin immediately available in
the users account after // executing the limit order. It does not include any future proceeds from the // maker portion which
will have withdrawn in the future cosmos . base . v1beta1 . Coin swap_out_coin =

3

[ ( gogoproto . moretags )

=
"yaml:\\"swap_out_coin\\"", ( gogoproto . nullable )

=
false , ( gogoproto . customtype )

=
"github.com/cosmos/cosmos-sdk/types.Coin" , ( gogoproto . jsontag )

=
"swap_out_coin" ] ; } Arguments

```

- QueryEstimatePlaceLimitOrderRequest
- : Request message for theEstimatePlaceLimitOrder
- query.* Creator
- - string (sdk.AccAddress): Account from which TokenIn is debited.
- - Receiver
- - string (sdk.AccAddress): Account to which TokenOut is credited or that will be allowed to withdraw or cancel a maker order.

- - TokenIn
- - (string): Token being “sold”.
- - TokenOut
- - (string): Token being “bought”.
- - TickIndex
- - (int64): Limit tick for a limit order, specified in terms of TokenIn to TokenOut.
- - AmountIn
- - (sdk.Int): Amount of TokenIn to be traded.
- - OrderType
- - (orderType): Type of limit order to be used. Must be one of: GOOD_TIL_CANCELLED, FILL_OR_KILL, IMMEDIATE_OR_CANCEL, JUST_IN_TIME, or GOOD_TIL_TIME.
- - ExpirationTime
- - (time.Time): Expiration time for order. Only valid for GOOD_TIL_TIME limit orders.

Curl Command (testnet):

```
curl https://rest-falcon.pion-1.ntrn.tech/neutron/dex/estimate_place_limit_order
```

PoolRequest

GET "/neutron/dex/pool/{pair_id}/{tick_index}/{fee}" Queries a pool by pair, tick and fee

Proto Messages

Request:

message

QueryPoolRequest

```
{ string pair_id =
1 ; int64 tick_index =
2 ; uint64 fee =
3 ; }
```

Response:

message

Pool

```
{ uint64 id =
1 ; PoolReserves lower_tick0 =
2 ; PoolReserves upper_tick1 =
3 ; }
```

message

QueryPoolResponse

```
{ Pool pool =
1
```

```
[ ( gogoproto . nullable )
```

=

true] ; } Arguments

- QueryPoolRequest
- : Request message for thePoolRequest
- query.* pairID
- - (string): The pair ID.
- - tickIndex
- - (int64): The tick index.
- - fee
- - (uint64): fee.

Curl Command (testnet):

curl https://rest-falcron.pion-1.ntrn.tech/neutron/dex/pool/ { pair_id } / { tick_index } / { fee }

PoolRequestByID

GET "/neutron/dex/pool/{pool_id}" Queries a pool by ID

Proto Messages

Request:

message

QueryPoolByIDRequest

{ uint64 pool_id =

1 ; } Response:

message

QueryPoolResponse

{ Pool pool =

1

[(gogoproto . nullable)

=

true] ; } Arguments

- QueryPoolByIDRequest
- : Request message for thePoolRequestByID
- query.* id
- - (uint64): Pool ID.

Curl Command (testnet):

curl https://rest-falcron.pion-1.ntrn.tech/neutron/dex/pool/ { pool_id }

GetPoolMetadata

GET "/neutron/dex/pool_metadata/{id}" Queries a PoolMetadata by ID

Proto Messages : Request:

message

QueryGetPoolMetadataRequest

```

{ uint64 id =
1 ; } Response:
message
PoolMetadata
{ uint64 id =
1 ; int64 tick =
2 ; uint64 fee =
3 ; PairID pair_id =
4 ; } message
PairID
{ string token0 =
1 ; string token1 =
2 ; }
message
QueryGetPoolMetadataResponse
{ PoolMetadata Pool_metadata =
1
[ ( gogoproto . nullable )
=
false ] ; } Arguments
    • QueryGetPoolMetadataRequest
    • : Request message for theGetPoolMetadata
    • query.* id
    •
      ◦ (uint64): Pool ID.

```

curl /neutron/dex/pool_metadata/ { id }

GetALLPoolMetadata

GET "/neutron/dex/pool_metadata" Queries a list of PoolMetadata items

Proto Messages :

Request:

message

QueryAllPoolMetadataRequest

```

{ cosmos . base . query . v1beta1 . PageRequest pagination =
1 ; } Response:

```

message

QueryAllPoolMetadataResponse

QueryAllPoolMetadataResponse

```

{ repeated

```

```

PoolMetadata pool_metadata =

```

```

1

```

[(gogoproto . nullable)

=

false] ; cosmos . base . query . v1beta1 . PageResponse pagination =

2 ; } Arguments

- QueryAllPoolMetadataRequest
- : Request message for theGetALLPoolMetadata
- query.* pagination
- - (cosmos.base.query.v1beta1.PageRequest): Pagination options.

Curl Command (testnet):

curl https://rest-falcron.pion-1.ntrn.tech/neutron/dex/pool_metadata [Previous Messages Next Overview](#)