

# Pools SDK

Pools are shared objects that represent a market. See [Query the Pool](#) for more information on pools.

The DeepBookV3 SDK exposes functions that you can call to read the state of a pool. These functions typically require a `managerKey`, `coinKey`, `poolKey`, or a combination of these. For details on these keys, see [DeepBookV3 SDK](#). The SDK includes some default keys that you can view in the `constants.ts` file.

... tip Decimal adjust all input quantities. All outputs are decimal adjusted. ...

Use `account` to retrieve the account information for a `BalanceManager` in a pool, which has the following form:

Parameters

Use `accountOpenOrders` to retrieve open orders for the balance manager and pool with the IDs you provide. The call returns a Promise that contains an array of open order IDs.

Parameters

Use `checkManagerBalance` to check the balance manager for a specific coin. The call returns a Promise in the form:

Parameters

Use `getOrder` to retrieve an order's information. The call returns a Promise in the `Order` struct, which has the following form:

Parameters

`poolKey` : String that identifies the pool to query. `orderId` : ID of the order to query.

Use `getQuoteQuantityOut` to retrieve the quote quantity out for the base quantity you provide. The call returns a Promise in the form:

where `deepRequired` is the amount of DEEP required for the dry run.

Parameters

Use `getBaseQuantityOut` to retrieve the base quantity out for the quote quantity that you provide. The call returns a Promise in the form:

where `deepRequired` is the amount of DEEP required for the dry run.

Parameters

Use `getQuantityOut` to retrieve the output quantities for the base or quote quantity you provide. You provide values for both quantities, but only one of them can be non-zero. The call returns a Promise with the form:

where `deepRequired` is the amount of DEEP required for the dry run.

Parameters

Use `getLevel2Range` to retrieve level 2 order book within the boundary price range you provide. The call returns a Promise in the form:

Parameters

Use `getLevel2TicksFromMid` to retrieve level 2 order book ticks from mid-price for a pool with the ID you provide. The call returns a Promise in the form:

Parameters

Use `lockedBalance` to retrieve a `BalanceManager` locked balance in the pool. The call returns a Promise in the `Order` struct, which has the following form:

Parameters

`poolKey` : String that identifies the pool to query. `balanceManagerKey` : key of the balance manager defined in the SDK.

Use `poolTradeParams` to retrieve the trade params for the pool, which has the following form:

Parameters

Use `vaultBalances` to get the vault balances for a pool with the ID you provide. The call returns a Promise in the form:

Parameters

Use `getPoolIdByAssets` to retrieve the pool ID for the asset types you provide. The call returns a Promise with the address of the pool if it's found.

Parameters

Use `midPrice` to retrieve the mid price for a pool with the ID that you provide. The call returns a Promise with the mid price.

Parameters

Use `whitelisted` to check if the pool with the ID you provide is whitelisted. The call returns a Promise as a boolean indicating whether the pool is whitelisted.

Parameters

## Pool functions

The DeepBookV3 SDK exposes functions that you can call to read the state of a pool. These functions typically require a `managerKey`, `coinKey`, `poolKey`, or a combination of these. For details on these keys, see [DeepBookV3 SDK](#). The SDK includes some default keys that you can view in the `constants.ts` file.

::: tip Decimal adjust all input quantities. All outputs are decimal adjusted. :::

Use `account` to retrieve the account information for a BalanceManager in a pool, which has the following form:

Parameters

Use `accountOpenOrders` to retrieve open orders for the balance manager and pool with the IDs you provide. The call returns a Promise that contains an array of open order IDs.

Parameters

Use `checkManagerBalance` to check the balance manager for a specific coin. The call returns a Promise in the form:

Parameters

Use `getOrder` to retrieve an order's information. The call returns a Promise in the Order struct, which has the following form:

Parameters

`poolKey` : String that identifies the pool to query. `orderId` : ID of the order to query.

Use `getQuoteQuantityOut` to retrieve the quote quantity out for the base quantity you provide. The call returns a Promise in the form:

where `deepRequired` is the amount of DEEP required for the dry run.

Parameters

Use `getBaseQuantityOut` to retrieve the base quantity out for the quote quantity that you provide. The call returns a Promise in the form:

where `deepRequired` is the amount of DEEP required for the dry run.

Parameters

Use `getQuantityOut` to retrieve the output quantities for the base or quote quantity you provide. You provide values for both quantities, but only one of them can be non-zero. The call returns a Promise with the form:

where `deepRequired` is the amount of DEEP required for the dry run.

#### Parameters

Use `getLevel2Range` to retrieve level 2 order book within the boundary price range you provide. The call returns a Promise in the form:

#### Parameters

Use `getLevel2TicksFromMid` to retrieve level 2 order book ticks from mid-price for a pool with the ID you provide. The call returns a Promise in the form:

#### Parameters

Use `lockedBalance` to retrieve a BalanceManager locked balance in the pool. The call returns a Promise in the Order struct, which has the following form:

#### Parameters

`poolKey` : String that identifies the pool to query. `balanceManagerKey` : key of the balance manager defined in the SDK.

Use `poolTradeParams` to retrieve the trade params for the pool, which has the following form:

#### Parameters

Use `vaultBalances` to get the vault balances for a pool with the ID you provide. The call returns a Promise in the form:

#### Parameters

Use `getPoolIdByAssets` to retrieve the pool ID for the asset types you provide. The call returns a Promise with the address of the pool if it's found.

#### Parameters

Use `midPrice` to retrieve the mid price for a pool with the ID that you provide. The call returns a Promise with the mid price.

#### Parameters

Use `whitelisted` to check if the pool with the ID you provide is whitelisted. The call returns a Promise as a boolean indicating whether the pool is whitelisted.

#### Parameters