Trade and Swap

DeepBook V2 is now deprecated. Update your integrations to use DeepBook V3 instead.

This section shows how to construct and execute a trade on the DeepBook protocol.

When you create a pool in DeepBook, you need to specify the BaseAsset , QuoteAsset , tick_size , and lot_size . You also need to pay a transaction fee (in SUI) to create a pool.

Function signature for create_pool in Move contract

TypeScript SDK for invoking create_pool

There are two types of orders, limit order and market order in DeepBook.

For limit order, you need to create a custodian account the first time you interact with a particular pool and deposit assets to the pool.

Tokens are created with different decimal precision. For example, SUI has nine decimals of precision, while the wormhole wrapped ethereum USDC has six decimals of precision. To accommodate proper pricing within DeepBook, every decimal is scaled with 10^9 precision. All the decimals in DeepBook meet the equation: 10^9 * (priceInRealWorld/priceInDeepBook) = (10^baseAssetDecimal) / (10^quoteAssetDecimal) .

Let's walk through an example between SUI/USDC. For this example, assume Sui price is $1.10 and is the base asset of the SUI/USDC trading pair.

Thus, the calculation is 10^9 * (1.10/ P_d) = 1000 , where P_d should be the price of sui in the DeepBook SUI/USDC trading pair. You can solve for P_d = 1100000 , so if the user wants to sell SUI at the market price in DeepBook, they should enter a price with six decimal points of precision.

There are four types of restrictions that you could put on limit orders.

NO_RESTRICTION : 0

Fill as much quantity as possible in the current transaction as taker, and inject the remaining as a maker order.

IMMEDIATE_OR_CANCEL : 1

Fill as much quantity as possible in the current transaction as taker, and cancel the rest of the order.

FILL_OR_KILL : 2

Only fill if the entire order size can be filled as taker in the current transaction. Otherwise, abort the entire transaction.

POST_OR_ABORT : 3

Only proceed if the entire order size can be posted to the order book as maker in the current transaction. Otherwise, abort the entire transaction.

Before placing a limit order, traders need to host a custodian account with asset deposited. Traders could generate a custodian account by invoking create_account to get an AccountCap object authorizing them to access their custodian accounts. The AccountCap object is the key to the custodian account, and if you transfer this object to others, the new owner has access to all the funds and orders in the custodian account.

Function signature for create_account in Move contract

TypeScript SDK for invoking create_account

Traders could deposit base/quote assets to their custodian account, so they could later use it for placing limit orders. We provide two functions to deposit base and quote assets, respectively.

Function signature for deposit base asset in Move contract

TypeScript SDK for invoking deposit_base

Function signature for deposit_quote in contract

TypeScript SDK for invoking deposit_quote

Ensure you have a custodian account with enough base/quote assets to trade. You can now place limit orders on DeepBook with the following functions:

Function signature for placing limit order in Move contract

TypeScript SDK for invoking place_limit_order

Placing a market order does not require a custodian account. The remaining order is canceled if it can be filled only partially.

Function signature for placing market order in Move contract:

TypeScript SDK for invoking place_market_order

Cancel a limit order placed onto the CLOB. Abort if order_id is invalid or if the order is not submitted by the transaction sender.

Function signature for canceling single order in Move contract

TypeScript SDK for invoking cancel_order

Cancel all limit orders under a certain account capacity.

Function signature for canceling single order in Move contract

TypeScript SDK for cancel all orders

Cancel multiple limit orders to save gas costs. Abort if any of the order ids is invalid or is not submitted by the sender. Note that this function can reduce gas costs even further if the caller has multiple orders at the same price level and if orders with the same price are grouped in the vector.

For example, if we have the following order_id to price mapping, {0: 100., 1: 200., 2: 100., 3: 200.} . Grouping order_ids like [0, 2, 1, 3] would make it the most gas efficient.

Function signature for batch cancel order

TypeScript SDK for cancel all orders

Create a pool

When you create a pool in DeepBook, you need to specify the BaseAsset , QuoteAsset , tick_size , and lot_size . You also need to pay a transaction fee (in SUI) to create a pool.

Function signature for create_pool in Move contract

TypeScript SDK for invoking create_pool

There are two types of orders, limit order and market order in DeepBook.

For limit order, you need to create a custodian account the first time you interact with a particular pool and deposit assets to the pool.

Tokens are created with different decimal precision. For example, SUI has nine decimals of precision, while the wormhole wrapped ethereum USDC has six decimals of precision. To accommodate proper pricing within DeepBook, every decimal is scaled with 10^9 precision. All the decimals in DeepBook meet the equation: 10^9 * (priceInRealWorld/priceInDeepBook) = (10^baseAssetDecimal) / (10^quoteAssetDecimal) .

Let's walk through an example between SUI/USDC. For this example, assume Sui price is $1.10 and is the base asset of the SUI/USDC trading pair.

Thus, the calculation is 10^9 * (1.10/ P_d) = 1000 , where P_d should be the price of sui in the DeepBook SUI/USDC trading pair. You can solve for P_d = 1100000 , so if the user wants to sell SUI at the market price in DeepBook, they should enter a price with six decimal points of precision.

There are four types of restrictions that you could put on limit orders.

NO_RESTRICTION : 0

Fill as much quantity as possible in the current transaction as taker, and inject the remaining as a maker order.

IMMEDIATE_OR_CANCEL : 1

Fill as much quantity as possible in the current transaction as taker, and cancel the rest of the order.

FILL_OR_KILL : 2

Only fill if the entire order size can be filled as taker in the current transaction. Otherwise, abort the entire transaction.

POST_OR_ABORT : 3

Only proceed if the entire order size can be posted to the order book as maker in the current transaction. Otherwise, abort the entire transaction.

Before placing a limit order, traders need to host a custodian account with asset deposited. Traders could generate a custodian account by invoking create_account to get an AccountCap object authorizing them to access their custodian accounts. The AccountCap object is the key to the custodian account, and if you transfer this object to others, the new owner has access to all the funds and orders in the custodian account.

Function signature for create_account in Move contract

TypeScript SDK for invoking create_account

Traders could deposit base/quote assets to their custodian account, so they could later use it for placing limit orders. We provide two functions to deposit base and quote assets, respectively.

Function signature for deposit base asset in Move contract

TypeScript SDK for invoking deposit_base

Function signature for deposit_quote in contract

TypeScript SDK for invoking deposit_quote

Ensure you have a custodian account with enough base/quote assets to trade. You can now place limit orders on DeepBook with the following functions:

Function signature for placing limit order in Move contract

TypeScript SDK for invoking place_limit_order

Placing a market order does not require a custodian account. The remaining order is canceled if it can be filled only partially.

Function signature for placing market order in Move contract:

TypeScript SDK for invoking place_market_order

Cancel a limit order placed onto the CLOB. Abort if order_id is invalid or if the order is not submitted by the transaction sender.

Function signature for canceling single order in Move contract

TypeScript SDK for invoking cancel_order

Cancel all limit orders under a certain account capacity.

Function signature for canceling single order in Move contract

TypeScript SDK for cancel all orders

Cancel multiple limit orders to save gas costs. Abort if any of the order ids is invalid or is not submitted by the sender. Note that this function can reduce gas costs even further if the caller has multiple orders at the same price level and if orders with the same price are grouped in the vector.

For example, if we have the following order_id to price mapping, {0: 100., 1: 200., 2: 100., 3: 200.} . Grouping order_ids like [0, 2, 1, 3] would make it the most gas efficient.

Function signature for batch cancel order

TypeScript SDK for cancel all orders

Create an order

There are two types of orders, limit order and market order in DeepBook.

For limit order, you need to create a custodian account the first time you interact with a particular pool and deposit assets to the pool.

Tokens are created with different decimal precision. For example, SUI has nine decimals of precision, while the wormhole wrapped ethereum USDC has six decimals of precision. To accommodate proper pricing within DeepBook, every decimal is scaled with 10^9 precision. All the decimals in DeepBook meet the equation: 10^9 * (priceInRealWorld/priceInDeepBook) = (10^baseAssetDecimal) / (10^quoteAssetDecimal) .

Let's walk through an example between SUI/USDC. For this example, assume Sui price is $1.10 and is the base asset of the SUI/USDC trading pair.

Thus, the calculation is 10^9 * (1.10/ P_d) = 1000 , where P_d should be the price of sui in the DeepBook SUI/USDC trading pair. You can solve for P_d = 1100000 , so if the user wants to sell SUI at the market price in DeepBook, they should enter a price with six decimal points of precision.

There are four types of restrictions that you could put on limit orders.

NO_RESTRICTION : 0

Fill as much quantity as possible in the current transaction as taker, and inject the remaining as a maker order.

IMMEDIATE_OR_CANCEL : 1

Fill as much quantity as possible in the current transaction as taker, and cancel the rest of the order.

FILL_OR_KILL : 2

Only fill if the entire order size can be filled as taker in the current transaction. Otherwise, abort the entire transaction.

POST_OR_ABORT : 3

Only proceed if the entire order size can be posted to the order book as maker in the current transaction. Otherwise, abort the entire transaction.

Before placing a limit order, traders need to host a custodian account with asset deposited. Traders could generate a custodian account by invoking create_account to get an AccountCap object authorizing them to access their custodian accounts. The AccountCap object is the key to the custodian account, and if you transfer this object to others, the new owner has access to all the funds and orders in the custodian account.

Function signature for create_account in Move contract

TypeScript SDK for invoking create_account

Traders could deposit base/quote assets to their custodian account, so they could later use it for placing limit orders. We provide two functions to deposit base and quote assets, respectively.

Function signature for deposit base asset in Move contract

TypeScript SDK for invoking deposit_base

Function signature for deposit_quote in contract

TypeScript SDK for invoking deposit_quote

Ensure you have a custodian account with enough base/quote assets to trade. You can now place limit orders on DeepBook with the following functions:

Function signature for placing limit order in Move contract

TypeScript SDK for invoking place_limit_order

Placing a market order does not require a custodian account. The remaining order is canceled if it can be filled only partially.

Function signature for placing market order in Move contract:

TypeScript SDK for invoking place_market_order

Cancel a limit order placed onto the CLOB. Abort if order_id is invalid or if the order is not submitted by the transaction sender.

Function signature for canceling single order in Move contract

TypeScript SDK for invoking cancel_order

Cancel all limit orders under a certain account capacity.

Function signature for canceling single order in Move contract

TypeScript SDK for cancel all orders

Cancel multiple limit orders to save gas costs. Abort if any of the order ids is invalid or is not submitted by the sender. Note that this function can reduce gas costs even further if the caller has multiple orders at the same price level and if orders with the same price are grouped in the vector.

For example, if we have the following order_id to price mapping, {0: 100., 1: 200., 2: 100., 3: 200.} . Grouping order_ids like [0, 2, 1, 3] would make it the most gas efficient.

Function signature for batch cancel order

TypeScript SDK for cancel all orders

Cancel order

Cancel a limit order placed onto the CLOB. Abort if order_id is invalid or if the order is not submitted by the transaction sender.

Function signature for canceling single order in Move contract

TypeScript SDK for invoking cancel_order

Cancel all limit orders under a certain account capacity.

Function signature for canceling single order in Move contract

TypeScript SDK for cancel all orders

Cancel multiple limit orders to save gas costs. Abort if any of the order ids is invalid or is not submitted by the sender. Note that this function can reduce gas costs even further if the caller has multiple orders at the same price level and if orders with the same price are grouped in the vector.

For example, if we have the following order_id to price mapping, {0: 100., 1: 200., 2: 100., 3: 200.} . Grouping order_ids like [0, 2, 1, 3] would make it the most gas efficient.

Function signature for batch cancel order

TypeScript SDK for cancel all orders