Address

The 32-byte address that is an account address (corresponding to a public key).

Objects owned by this address, optionally filter -ed.

Total balance of all coins with marker type owned by this address. If type is not supplied, it defaults to 0x2::sui::SUI .

The balances of all coin types owned by this address.

The coin objects for this address.

type is a filter on the coin's type parameter, defaulting to 0x2::sui::SUI .

The 0x3::staking_pool::StakedSui objects owned by this address.

The domain explicitly configured as the default domain pointing to this address.

The SuinsRegistration NFTs owned by this address. These grant the owner the capability to manage the associated domain.

Similar behavior to the transactionBlocks in Query but supporting the additional AddressTransactionBlockRelationship filter, which defaults to SENT .

scanLimit restricts the number of candidate transactions scanned when gathering a page of results. It is required for queries that apply more than two complex filters (on function, kind, sender, recipient, input object, changed object, or ids), and can be at most serviceConfig.maxScanLimit .

When the scan limit is reached the page will be returned even if it has fewer than first results when paginating forward (last when paginating backwards). If there are more transactions to scan, pageInfo.hasNextPage (or pageInfo.hasPreviousPage) will be set to true , and PageInfo.endCursor (or PageInfo.startCursor) will be set to the last transaction that was scanned as opposed to the last (or first) transaction in the page.

Requesting the next (or previous) page after this cursor will resume the search, scanning the next scanLimit many transactions in the direction of pagination, and so on until all transactions in the scanning range have been visited.

By default, the scanning range includes all transactions known to GraphQL, but it can be restricted by the after and before cursors, and the beforeCheckpoint , afterCheckpoint and atCheckpoint filters.

Interface implemented by GraphQL types representing entities that can own objects. Object owners are identified by an address which can represent either the public key of an account or another object. The same address can only refer to an account or an object, never both, but it is not possible to know which up-front.

address query ● resolveSuinsAddress query

AddressConnection object ● AddressEdge object ● Event object ● GasInput object ● Owner object ● TransactionBlock object ● Validator object

Authenticator union