Address

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

```
type Address implements IOwner {
  address: SuiAddress!
  objects(
    first: Int
    after: String
    last: Int
    before: String
    filter: ObjectFilter
  ): MoveObjectConnection!
  balance(
    type: String
  ): Balance
  balances(
    first: Int
    after: String
    last: Int
    before: String
  ): BalanceConnection!
  coins(
    first: Int
    after: String
    last: Int
    before: String
    type: String
  ): CoinConnection!
  stakedSuis(
    first: Int
    after: String
    last: Int
    before: String
  ): StakedSuiConnection!
  defaultSuinsName(
    format: DomainFormat
  ): String
  suinsRegistrations(
    first: Int
    after: String
    last: Int
    before: String
  ): SuinsRegistrationConnection!
  transactionBlocks(
    first: Int
    after: String
    last: Int
    before: String
    relation: AddressTransactionBlockRelationship
    filter: TransactionBlockFilter
    scanLimit: Int
  ): TransactionBlockConnection!
}
```

Fields

`Address.address` ● `SuiAddress!` non-null scalar

`Address.objects` ● `MoveObjectConnection!` non-null object

Objects owned by this address, optionally `filter`-ed.

`Address.objects.first` ● `Int` scalar

`Address.objects.after` ● `String` scalar

`Address.objects.last` ● `Int` scalar

`Address.objects.before` ● `String` scalar

`Address.objects.filter` ● `ObjectFilter` input

`Address.balance` ● `Balance` object

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

`Address.balance.type` ● `String` scalar

`Address.balances` ● `BalanceConnection!` non-null object

The balances of all coin types owned by this address.

`Address.balances.first` ● `Int` scalar

`Address.balances.after` ● `String` scalar

`Address.balances.last` ● `Int` scalar

`Address.balances.before` ● `String` scalar

`Address.coins` ● `CoinConnection!` non-null object

The coin objects for this address.

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

`Address.coins.first` • `Int` scalar

`Address.coins.after` • `String` scalar

`Address.coins.last` • `Int` scalar

`Address.coins.before` • `String` scalar

`Address.coins.type` • `String` scalar

`Address.stakedSuis` • `StakedSuiConnection!` non-null object

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

`Address.stakedSuis.first` • `Int` scalar

`Address.stakedSuis.after` • `String` scalar

`Address.stakedSuis.last` • `Int` scalar

`Address.stakedSuis.before` • `String` scalar

`Address.defaultSuinsName` • `String` scalar

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

`Address.defaultSuinsName.format` • `DomainFormat` enum

`Address.suinsRegistrations` • `SuinsRegistrationConnection!` non-null object

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

`Address.suinsRegistrations.first` • `Int` scalar

`Address.suinsRegistrations.after` • `String` scalar

`Address.suinsRegistrations.last` • `Int` scalar

`Address.suinsRegistrations.before` • `String` scalar

`Address.transactionBlocks` • `TransactionBlockConnection!` non-null object

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.

`Address.transactionBlocks.first` . `Int` scalar

`Address.transactionBlocks.after` . `String` scalar

`Address.transactionBlocks.last` . `Int` scalar

`Address.transactionBlocks.before` . `String` scalar

`Address.transactionBlocks.relation` . `AddressTransactionBlockRelationship` enum

`Address.transactionBlocks.filter` . `TransactionBlockFilter` input

`Address.transactionBlocks.scanLimit` . `Int` scalar

Interfaces

`IOwner` interface

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.

Returned By

`address` query . `resolveSuinsAddress` query

Member Of

`AddressConnection` object . `AddressEdge` object . `Event` object . `GasInput` object . `Owner` object . `TransactionBlock` object . `Validator` object

Implemented By

`Authenticator` union

Edit this page