MovePackage

A MovePackage is a kind of Move object that represents code that has been published on chain. It exposes information about its modules, type definitions, functions, and dependencies.

```
type MovePackage implements IObject, 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!
  version: UInt53!
  status: ObjectKind!
  digest: String
```

```
  owner: ObjectOwner
  previousTransactionBlock: TransactionBlock
  storageRebate: BigInt
  receivedTransactionBlocks(
    first: Int
    after: String
    last: Int
    before: String
    filter: TransactionBlockFilter
    scanLimit: Int
  ): TransactionBlockConnection!
  bcs: Base64
  packageAtVersion(
    version: Int!
  ): MovePackage
  packageVersions(
    first: Int
    after: String
    last: Int
    before: String
    filter: MovePackageVersionFilter
  ): MovePackageConnection!
  latestPackage: MovePackage!
  module(
    name: String!
  ): MoveModule
  modules(
    first: Int
    after: String
    last: Int
    before: String
  ): MoveModuleConnection
  linkage: [Linkage!]
  typeOrigins: [TypeOrigin!]
  packageBcs: Base64
  moduleBcs: Base64
}
```

Fields

`MovePackage.address.SuiAddress!` non-null scalar

`MovePackage.objects.MoveObjectConnection!` non-null object

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

Note that objects owned by a package are inaccessible, because packages are immutable and cannot be owned by an address.

`MovePackage.objects.first.Int` scalar

MovePackage.objects.after.String scalar

MovePackage.objects.last.Int scalar

MovePackage.objects.before.String scalar

MovePackage.objects.filter.ObjectFilter input

MovePackage.balance.Balance object

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

Note that coins owned by a package are inaccessible, because packages are immutable and cannot be owned by an address.

MovePackage.balance.type.String scalar

MovePackage.balances.BalanceConnection! non-null object

The balances of all coin types owned by this package.

Note that coins owned by a package are inaccessible, because packages are immutable and cannot be owned by an address.

MovePackage.balances.first.Int scalar

MovePackage.balances.after.String scalar

MovePackage.balances.last.Int scalar

MovePackage.balances.before.String scalar

MovePackage.coins.CoinConnection! non-null object

The coin objects owned by this package.

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

Note that coins owned by a package are inaccessible, because packages are immutable and cannot be owned by an address.

MovePackage.coins.first.Int scalar

MovePackage.coins.after.String scalar

MovePackage.coins.last.Int scalar

MovePackage.coins.before.String scalar

MovePackage.coins.type.String scalar

MovePackage.stakedSuis.StakedSuiConnection! non-null object

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

Note that objects owned by a package are inaccessible, because packages are immutable and cannot be owned by an address.

MovePackage.stakedSuis.first.Int scalar

MovePackage.stakedSuis.after.String scalar

MovePackage.stakedSuis.last.Int scalar

MovePackage.stakedSuis.before.String scalar

MovePackage.defaultSuinsName.String scalar

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

MovePackage.defaultSuinsName.format.DomainFormat enum

MovePackage.suinsRegistrations.SuinsRegistrationConnection! non-null object

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

Note that objects owned by a package are inaccessible, because packages are immutable and cannot be owned by an address.

MovePackage.suinsRegistrations.first.Int scalar

MovePackage.suinsRegistrations.after.String scalar

MovePackage.suinsRegistrations.last.Int scalar

MovePackage.suinsRegistrations.before.String scalar

MovePackage.version.UInt53! non-null scalar

MovePackage.status.ObjectKind! non-null enum

The current status of the object as read from the off-chain store. The possible states are: NOT_INDEXED, the object is loaded from serialized data, such as the contents of a genesis or system package upgrade transaction. LIVE, the version returned is the most recent for the object, and it is not deleted or wrapped at that version. HISTORICAL, the object was referenced at a specific

version or checkpoint, so is fetched from historical tables and may not be the latest version of the object. WRAPPED_OR_DELETED, the object is deleted or wrapped and only partial information can be loaded."

`MovePackage.digest`.`String` **scalar**

32-byte hash that identifies the package's contents, encoded as a Base58 string.

`MovePackage.owner`.`ObjectOwner` **union**

The owner type of this object: Immutable, Shared, Parent, Address Packages are always Immutable.

`MovePackage.previousTransactionBlock`.`TransactionBlock` **object**

The transaction block that published or upgraded this package.

`MovePackage.storageRebate`.`BigInt` **scalar**

The amount of SUI we would rebate if this object gets deleted or mutated. This number is recalculated based on the present storage gas price.

Note that packages cannot be deleted or mutated, so this number is provided purely for reference.

`MovePackage.receivedTransactionBlocks`.`TransactionBlockConnection!` **non-null** **object**

The transaction blocks that sent objects to this package.

Note that objects that have been sent to a package become inaccessible.

`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.

MovePackage.receivedTransactionBlocks.first.Int scalar

MovePackage.receivedTransactionBlocks.after.String scalar

MovePackage.receivedTransactionBlocks.last.Int scalar

MovePackage.receivedTransactionBlocks.before.String scalar

MovePackage.receivedTransactionBlocks.filter.TransactionBlockFilter input

MovePackage.receivedTransactionBlocks.scanLimit.Int scalar

MovePackage.bcs.Base64 scalar

The Base64-encoded BCS serialization of the package's content.

MovePackage.packageAtVersion.MovePackage object

Fetch another version of this package (the package that shares this package's original ID, but has the specified `version`).

MovePackage.packageAtVersion.version.Int! non-null scalar

MovePackage.packageVersions.MovePackageConnection! non-null object

Fetch all versions of this package (packages that share this package's original ID), optionally bounding the versions exclusively from below with `afterVersion`, or from above with `beforeVersion`.

MovePackage.packageVersions.first.Int scalar

MovePackage.packageVersions.after.String scalar

MovePackage.packageVersions.last.Int scalar

MovePackage.packageVersions.before.String scalar

MovePackage.packageVersions.filter.MovePackageVersionFilter input

MovePackage.latestPackage.MovePackage! non-null object

Fetch the latest version of this package (the package with the highest `version` that shares this packages's original ID)

MovePackage.module.MoveModule object

A representation of the module called `name` in this package, including the structs and functions it defines.

`MovePackage.module.name.String!` non-null scalar

`MovePackage.modules.MoveModuleConnection` object

Paginate through the MoveModules defined in this package.

`MovePackage.modules.first.Int` scalar

`MovePackage.modules.after.String` scalar

`MovePackage.modules.last.Int` scalar

`MovePackage.modules.before.String` scalar

`MovePackage.linkage.[Linkage!]` list object

The transitive dependencies of this package.

`MovePackage.typeOrigins.[TypeOrigin!]` list object

The (previous) versions of this package that introduced its types.

`MovePackage.packageBcs.Base64` scalar

BCS representation of the package itself, as a MovePackage.

`MovePackage.moduleBcs.Base64` scalar

BCS representation of the package's modules. Modules appear as a sequence of pairs (module name, followed by module bytes), in alphabetic order by module name.

Interfaces

`IObject` interface

Interface implemented by on-chain values that are addressable by an ID (also referred to as its address). This includes Move objects and packages.

`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

latestPackage query . package query . packageByName query

Member Of

MoveModule object . MovePackage object . MovePackageConnection object . MovePackageEdge object . Object object

Edit this page