Sui Full Node gRPC

BETA FEATURE

This content describes a beta feature or service. Beta features and services are in active development, so details are likely to change.

This feature or service is currently available in

- **Devnet**
- **Testnet**
- **Mainnet**

Sui Full node gRPC API will replace the JSON-RPC on Full nodes, such that JSON-RPC will be deprecated when gRPC API is generally available.

sui.node.v2.proto

The sui.node.v2 package contains API definitions for services that are expected to run on Full nodes.

BalanceChange

The delta, or change, in balance for an address for a particular `Coin` type.

Fields

One of

address

Type | Address Label | optional

Description | The account address that is affected by this balance change event.

amount

Type | I128 Label | optional

Description | The amount or change in balance.

coin_type

Type | TypeTag Label | optional

Description | The `Coin` type of this balance change event.

BalanceChanges

Set of `BalanceChange`s that occurred as the result of a transaction.

This set of events are calculated by analyzing all input and output `Coin` type objects.

Fields

balance_changes

Type | BalanceChange Label | repeated

EffectsFinality

Indicates the finality of the executed transaction.

Fields

One of

certified

Type | ValidatorAggregatedSignature

Description | A quorum certificate certifying that a transaction is final but might not be included in a checkpoint yet.

checkpointed

Type | uint64

Description | Sequence number of the checkpoint that includes the transaction.

quorum_executed

Type | Empty

Description | Indicates that a quorum of validators has executed the transaction but that it might not be included in a checkpoint yet.

ExecuteTransactionOptions

Fields

One of

balance_changes

Type | bool Label | optional

Description | Include the `BalanceChanges` in the response. Defaults to `false` if not included.

effects

Type | bool Label | optional

Description | Include the `sui.types.TransactionEffects` message in the response. Defaults to `false` if not included.

effects_bcs

Type | bool Label | optional

Description | Include the `TransactionEffects` formatted as BCS in the response. Defaults to `false` if not included.

events

Type | bool Label | optional

Description | Include the `sui.types.TransactionEvents` message in the response. Defaults to `false` if not included.

events_bcs

Type | bool Label | optional

Description | Include the `TransactionEvents` formatted as BCS in the response. Defaults to `false` if not included.

ExecuteTransactionRequest

Request message for `NodeService.ExecuteTransaction`.

Note: You must provide only one of `transaction` or `transaction_bcs`.

Fields

One of

options

Type | ExecuteTransactionOptions Label | optional

Description | Optional. Options for specifying which parts of the `ExecuteTransactionResponse` should be returned.

signatures

Type | UserSignatures Label | optional

Description | Optional. Set of `UserSiganture`s authorizing the execution of the provided transaction.

signatures_bytes

Type | UserSignaturesBytes Label | optional

Description | Optional. Set of `UserSiganture`s authorizing the execution of the provided transaction, encoded as bytes.

transaction

Type | Transaction Label | optional

Description | Optional. The transaction to execute.

transaction_bcs

Type | Bcs Label | optional

Description | Optional. The transaction to execute, encoded as BCS bytes.

ExecuteTransactionResponse

Response message for `NodeService.ExecuteTransaction`.

Fields

One of

balance_changes

Type | BalanceChanges Label | optional

Description | Optional. Set of balance change events as a result of this transaction.

effects

Type | TransactionEffects Label | optional

Description | Optional. The `TransactionEffects` for this transaction.

effects_bcs

Type | Bcs Label | optional

Description | Optional. The TransactionEffects for this transaction encoded as BCS bytes.

events

Type | TransactionEvents Label | optional

Description | Optional. The `TransactionEvents` for this transaction. This field might be empty, even if it was explicitly requested, if the transaction didn't produce any events. `sui.types.TransactionEffects.events_digest` is populated if the transaction produced any events.

events_bcs

Type | Bcs Label | optional

Description | Optional. The TransactionEvents for this transaction encoded as BCS bytes.

finality

Type | EffectsFinality Label | optional

Description | Indicates the finality of the executed transaction.

FullCheckpointObject

An object used by or produced from a transaction.

Fields

One of

digest

Type | Digest Label | optional

Description | The digest of this object.

object

Type | Object Label | optional

Description | Optional. The object itself.

object_bcs

Type | Bcs Label | optional

Description | Optional. The object encoded as BCS bytes.

object_id

Type | ObjectId Label | optional

Description | The `ObjectId` of this object.

version

Type | uint64 Label | optional

Description | The version of this object.

FullCheckpointObjects

Set of objects used by or produced from a transaction.

Fields

objects

Type | FullCheckpointObject Label | repeated

FullCheckpointTransaction

A transaction, with all of its inputs and outputs.

Fields

One of

digest

Type | Digest Label | optional

Description | The digest of this transaction.

effects

Type | TransactionEffects Label | optional

Description | Optional. The `TransactionEffects` for this transaction.

effects_bcs

Type | Bcs Label | optional

Description | Optional. The TransactionEffects for this transaction encoded as BCS bytes.

events

Type | TransactionEvents Label | optional

Description | Optional. The `TransactionEvents` for this transaction. This field might be empty, even if it was explicitly requested, if the transaction didn't produce any events. `sui.types.TransactionEffects.events_digest` is populated if the transaction produced any events.

events_bcs

Type | Bcs Label | optional

Description | Optional. The TransactionEvents for this transaction encoded as BCS bytes.

input_objects

Type | FullCheckpointObjects Label | optional

Description | Optional. Set of input objects used during the execution of this transaction.

output_objects

Type | FullCheckpointObjects Label | optional

Description | Optional. Set of output objects produced from the execution of this transaction.

transaction

Type | Transaction Label | optional

Description | Optional. The transaction itself.

transaction_bcs

Type | Bcs Label | optional

Description | Optional. The Transaction encoded as BCS bytes.

GetCheckpointOptions

Options for which parts of the `GetCheckpointResponse` should be returned.

Fields

One of

contents

Type | bool Label | optional

Description | Include the `sui.types.CheckpointContents` message in the response. Defaults to `false` if not included.

contents_bcs

Type | bool Label | optional

Description | Include the `CheckpointContents` formatted as BCS in the response. Defaults to `false` if not included.

signature

Type | bool Label | optional

Description | Include the `sui.types.ValidatorAggregatedSignature` in the response. Defaults to `false` if not included.

summary

Type | bool Label | optional

Description | Include the `sui.types.CheckpointSummary` in the response. Defaults to `false` if not included.

summary_bcs

Type | bool Label | optional

Description | Include the `CheckpointSummary` formatted as BCS in the response. Defaults to `false` if not included.

GetCheckpointRequest

Request message for `NodeService.GetCheckpoint`.

At most, provide one of `sequence_number` or `digest`. An error is returned if you attempt to provide both. If you provide neither, the service returns the latest executed checkpoint.

Fields

One of

digest

Type | Digest Label | optional

Description | Optional. The digest of the requested checkpoint.

options

Type | GetCheckpointOptions Label | optional

Description | Optional. Options for specifying which parts of the `GetCheckpointResponse` should be returned.

sequence_number

Type | uint64 Label | optional

Description | Optional. The sequence number of the requested checkpoint.

GetCheckpointResponse

Response message for `NodeService.GetCheckpoint`.

Fields

One of

contents

Type | CheckpointContents Label | optional

Description | Optional. The `CheckpointContents` for this checkpoint.

contents_bcs

Type | Bcs Label | optional

Description | Optional. The CheckpointContents for this checkpoint encoded as BCS bytes.

digest

Type | Digest Label | optional

Description | The digest of this checkpoint's `CheckpointSummary`.

sequence_number

Type | uint64 Label | optional

Description | The sequence number of this checkpoint.

signature

Type | ValidatorAggregatedSignature Label | optional

Description | Optional. An aggregated quorum signature from the validator committee that certifies this checkpoint.

summary

Type | CheckpointSummary Label | optional

Description | Optional. The `CheckpointSummary` for this checkpoint.

summary_bcs

Type | Bcs Label | optional

Description | Optional. The CheckpointSummary for this checkpoint encoded as BCS bytes.

GetCommitteeRequest

Request message for NodeService.GetCommittee.

Fields

One of

epoch

Type | uint64 Label | optional

Description | Optional. Request the sui.types.ValidatorCommittee corresponding to the provided epoch. If no epoch is provided the committee for the current epoch will be returned.

GetCommitteeResponse

Response message for `NodeService.GetCommittee`.

Fields

One of

committee

Type | ValidatorCommittee Label | optional

Description | The committee of either the requested epoch or the current epoch.

GetFullCheckpointOptions

Options for which parts of the `GetFullCheckpointResponse` should be returned.

Fields

One of

contents

Type | bool Label | optional

Description | Include the `sui.types.CheckpointContents` message in the response. Defaults to `false` if not included.

contents_bcs

Type | bool Label | optional

Description | Include the `CheckpointContents` formatted as BCS in the response. Defaults to `false` if not included.

effects

Type | bool Label | optional

Description | Include the `sui.types.TransactionEffects` message in the response. Defaults to `false` if not included.

effects_bcs

Type | bool Label | optional

Description | Include the `TransactionEffects` formatted as BCS in the response. Defaults to `false` if not included.

events

Type | bool Label | optional

Description | Include the `sui.types.TransactionEvents` message in the response. Defaults to `false` if not included.

events_bcs

Type | bool Label | optional

Description | Include the `TransactionEvents` formatted as BCS in the response. Defaults to `false` if not included.

input_objects

Type | bool Label | optional

Description | Include the input objects for transactions in the response. Defaults to `false` if not included.

object

Type | bool Label | optional

Description | Include the `sui.types.Object` message in the response. Defaults to `false` if not included.

object_bcs

Type | bool Label | optional

Description | Include the object formatted as BCS in the response. Defaults to `false` if not included.

output_objects

Type | bool Label | optional

Description | Include the output objects for transactions in the response. Defaults to `false` if not included.

signature

Type | bool Label | optional

Description | Include the `sui.types.ValidatorAggregatedSignature` in the response. Defaults to `false` if not included.

summary

Type | bool Label | optional

Description | Include the `sui.types.CheckpointSummary` in the response. Defaults to `false` if not included.

summary_bcs

Type | bool Label | optional

Description | Include the `CheckpointSummary` formatted as BCS in the response. Defaults to `false` if not included.

transaction

Type | bool Label | optional

Description | Include the `sui.types.Transaction` message in the response. Defaults to `false` if not included.

transaction_bcs

Type | bool Label | optional

Description | Include the transaction formatted as BCS in the response. Defaults to `false` if not

included.

GetFullCheckpointRequest

Request message for `NodeService.GetFullCheckpoint`.

At most, provide one of `sequence_number` or `digest`. An error is returned if you provide both. If you provide neither, the service returns the latest executed checkpoint.

Fields

One of

digest

Type | Digest Label | optional

Description | Optional. The digest of the requested checkpoint.

options

Type | GetFullCheckpointOptions Label | optional

Description | Optional. Options for specifying which parts of the `GetFullCheckpointResponse` should be returned.

sequence_number

Type | uint64 Label | optional

Description | Optional. The sequence number of the requested checkpoint.

GetFullCheckpointResponse

Response message for `NodeService.GetFullCheckpoint`.

Fields

transactions

Type | FullCheckpointTransaction Label | repeated

Description | List of transactions included in this checkpoint.

One of

contents

Type | CheckpointContents Label | optional

Description | Optional. The `CheckpointContents` for this checkpoint.

contents_bcs

Type | Bcs Label | optional

Description | Optional. The CheckpointContents for this checkpoint encoded as BCS bytes.

digest

Type | Digest Label | optional

Description | The digest of this checkpoint's `CheckpointSummary`.

sequence_number

Type | uint64 Label | optional

Description | The sequence number of this checkpoint.

signature

Type | ValidatorAggregatedSignature Label | optional

Description | Optional. An aggregated quorum signature from the validator committee that certifies this checkpoint.

summary

Type | CheckpointSummary Label | optional

Description | Optional. The `CheckpointSummary` for this checkpoint.

summary_bcs

Type | Bcs Label | optional

Description | Optional. The CheckpointSummary for this checkpoint encoded as BCS bytes.

GetNodeInfoRequest

Request message for `NodeService.GetNodeInfo`.

GetNodeInfoResponse

Response message for `NodeService.GetNodeInfo`.

Fields

One of

chain

Type | string Label | optional

Description | Human-readable name of the chain that this node is on. This is intended to be a human-readable name like `mainnet`, `testnet`, and so on.

chain_id

Type | Digest Label | optional

Description | The chain identifier of the chain that this node is on. The chain identifier is the digest of the genesis checkpoint, the checkpoint with sequence number 0.

checkpoint_height

Type | uint64 Label | optional

Description | Checkpoint height of the most recently executed checkpoint.

epoch

Type | uint64 Label | optional

Description | Current epoch of the node based on its highest executed checkpoint.

lowest_available_checkpoint

Type | uint64 Label | optional

Description | The lowest checkpoint for which checkpoints and transaction data are available.

lowest_available_checkpoint_objects

Type | uint64 Label | optional

Description | The lowest checkpoint for which object data is available.

software_version

Type | string Label | optional

Description | Software version of the `sui-node` binary.

timestamp

Type | Timestamp Label | optional

Description | Unix timestamp of the most recently executed checkpoint.

GetObjectOptions

Fields

One of

object

Type | bool Label | optional

Description | Include the `sui.types.Object` message in the response. Defaults to `false` if not included.

object_bcs

Type | bool Label | optional

Description | Include the object formatted as BCS in the response. Defaults to `false` if not included.

GetObjectRequest

Request message for `NodeService.GetObject`.

Fields

One of

object_id

Type | ObjectId Label | optional

Description | Required. The `ObjectId` of the requested object.

options

Type | GetObjectOptions Label | optional

Description | Optional. Options for specifying which parts of the `GetObjectResponse` should be returned.

version

Type | uint64 Label | optional

Description | Optional. Request that a specific version of the requested object is returned. If no version is provided, then then the latest version for the object is returned.

GetObjectResponse

Response message for `NodeService.GetObject`.

Fields

One of

digest

Type | Digest Label | optional

Description | The digest of this object.

object

Type | Object Label | optional

Description | Optional. The object itself.

object_bcs

Type | Bcs Label | optional

Description | Optional. The Object encoded as BCS bytes.

object_id

Type | ObjectId Label | optional

Description | The `ObjectId` of this object.

version

Type | uint64 Label | optional

Description | The version of this object.

GetTransactionOptions

Options for which parts of the `GetTransactionResponse` should be returned.

Fields

One of

effects

Type | bool Label | optional

Description | Include the `sui.types.TransactionEffects` message in the response. Defaults to `false` if not included.

effects_bcs

Type | bool Label | optional

Description | Include the `TransactionEffects` formatted as BCS in the response. Defaults to `false` if not included.

events

Type | bool Label | optional

Description | Include the `sui.types.TransactionEvents` message in the response. Defaults to `false` if not included.

events_bcs

Type | bool Label | optional

Description | Include the `TransactionEvents` formatted as BCS in the response. Defaults to `false` if not included.

signatures

Type | bool Label | optional

Description | Include the set of `sui.types.UserSignature`s in the response. Defaults to `false` if not included.

signatures_bytes

Type | bool Label | optional

Description | Include the set of `UserSignature`s encoded as bytes in the response. Defaults to `false` if not included.

transaction

Type | bool Label | optional

Description | Include the `sui.types.Transaction` message in the response. Defaults to `false` if not included.

transaction_bcs

Type | bool Label | optional

Description | Include the transaction formatted as BCS in the response. Defaults to `false` if not included.

GetTransactionRequest

Request message for `NodeService.GetTransaction`.

Fields

One of

digest

Type | Digest Label | optional

Description | Required. The digest of the requested transaction.

options

Type | GetTransactionOptions Label | optional

Description | Optional. Options for specifying which parts of the `GetTransactionResponse` should be returned.

GetTransactionResponse

Response message for `NodeService.GetTransactio`n.

Fields

One of

checkpoint

Type | uint64 Label | optional

Description | The sequence number for the checkpoint that includes this transaction.

digest

Type | Digest Label | optional

Description | The digest of this Transaction.

effects

Type | TransactionEffects Label | optional

Description | Optional. The `TransactionEffects` for this transaction.

effects_bcs

Type | Bcs Label | optional

Description | Optional. The TransactionEffects for this transaction encoded as BCS bytes.

events

Type | TransactionEvents Label | optional

Description | Optional. The `TransactionEvents` for this transaction. This field might be empty, even if it was explicitly requested, if the transaction didn't produce any events. `sui.types.TransactionEffects.events_digest` is populated if the transaction produced any events.

events_bcs

Type | Bcs Label | optional

Description | Optional. The TransactionEvents for this transaction encoded as BCS bytes.

signatures

Type | UserSignatures Label | optional

Description | Optional. List of user signatures that are used to authorize the execution of this transaction.

signatures_bytes

Type | UserSignaturesBytes Label | optional

Description | Optional. List of UserSignatures encoded as bytes.

timestamp

Type | Timestamp Label | optional

Description | The Unix timestamp of the checkpoint that includes this transaction.

transaction

Type | Transaction Label | optional

Description | Optional. The transaction itself.

transaction_bcs

Type | Bcs Label | optional

Description | Optional. The Transaction encoded as BCS bytes.

UserSignatures

List of `UserSignature`s used to authorize a transaction.

Fields

signatures

Type | UserSignature Label | repeated

UserSignaturesBytes

List of `UserSignature`s used to authorize a transaction encoded as bytes.

Fields

signatures

Type | bytes Label | repeated

sui.types.proto

Protobuf definitions of public Sui core types.

This file contains a complete set of protobuf definitions for all of the public sui core types. All sui types are intended to have a 1:1 mapping to a protobuf message defined in this file and be able to roundtrip to/from their rust and protobuf definitions assuming a sufficiently up-to-date version of both these definitions.

For more information on the types these proto messages correspond with, see the documentation for their rust versions defined in the `sui-sdk-types` library.

Use of `optional`

These message definitions use protobuf version 3 (proto3). In proto3, fields that are primitives (that is, they are not a `message`) and are not present on the wire are zero-initialized. To gain the ability to detect field presence, these definitions follow the convention of having all fields marked `optional`, and wrapping `repeated` fields in a message as needed.

Even if a field is marked as `optional`, it might not actually be optional from the perspective of the Sui protocol. Such fields are explicitly labled as `Required` or `Optional` in their documentation.

ActiveJwk

A new JWK.

Fields

One of

epoch

Type | uint64 Label | optional

Description | Most recent epoch in which the JWK was validated.

id

Type | JwkId Label | optional

Description | Identifier used to uniquely identify a JWK.

jwk

Type | Jwk Label | optional

Description | The JWK.

Address

Unique identifier for an account on the Sui blockchain.

An `Address` is a 32-byte pseudonymous identifier used to uniquely identify an account and asset-ownership on the Sui blockchain. Often, human-readable addresses are encoded in hexadecimal with a `0x` prefix. For example, this is a valid Sui address: `0x02a212de6a9dfa3a69e22387acfbafbb1a9e591bd9d636e7895dcfc8de05f331`.

Fields

One of

address

Type | bytes Label | optional

Description | Required. 32-byte address.

AddressDeniedForCoinError

Address is denied for this coin type.

Fields

One of

address

Type | Address Label | optional

Description | Required. Denied address.

coin_type

Type | string Label | optional

Description | Required. Coin type.

Argument

An argument to a programmable transaction command.

Fields

One of

gas

Type | Empty

Description | The gas coin. The gas coin can only be used by-ref, except for with `TransferObjects`, which can use it by-value.

input

Type | uint32

Description | One of the input objects or primitive values (from `ProgrammableTransaction` inputs).

nested_result

Type | NestedResult

Description | Like a `Result` but it accesses a nested result. Currently, the only usage of this is to access a value from a Move call with multiple return values.

result

Type | uint32

Description | The result of another command (from `ProgrammableTransaction` commands).

AuthenticatorStateExpire

Expire old JWKs.

Fields

One of

authenticator_object_initial_shared_version

Type | uint64 Label | optional

Description | The initial version of the authenticator object that it was shared at.

min_epoch

Type | uint64 Label | optional

Description | Expire JWKs that have a lower epoch than this.

AuthenticatorStateUpdate

Update the set of valid JWKs.

Fields

- **new_active_jwks**

  Type | ActiveJwk Label | repeated

  Description | Newly active JWKs.

- **One of**

  - **authenticator_object_initial_shared_version**

    Type | uint64 Label | optional

    Description | The initial version of the authenticator object that it was shared at.

  - **epoch**

    Type | uint64 Label | optional

    Description | Epoch of the authenticator state update transaction.

  - **round**

    Type | uint64 Label | optional

    Description | Consensus round of the authenticator state update.

Bcs

Message that represents a type that is serialized and encoded using the BCS format.

Fields

- **One of**

  - **bcs**

    Type | bytes Label | optional

    Description | Required. Bytes of a BCS encoded value.

Bn254FieldElement

A point on the BN254 elliptic curve.

Fields

One of

element

Type | bytes Label | optional

Description | Required. 32-byte big-endian field element.

CancelledTransaction

A transaction that was cancelled.

Fields

version_assignments

Type | VersionAssignment Label | repeated

Description | List of object version assignments.

One of

digest

Type | Digest Label | optional

Description | Digest of the cancelled transaction.

CancelledTransactions

Set of cancelled transactions.

Fields

cancelled_transactions

Type | CancelledTransaction Label | repeated

ChangeEpoch

System transaction used to change the epoch.

Fields

system_packages

Type | SystemPackage Label | repeated

Description | System packages (specifically framework and Move stdlib) that are written before the new epoch starts. This tracks framework upgrades on chain. When executing the `ChangeEpoch` txn, the validator must write out the following modules. Modules are provided with the version they will be upgraded to, their modules in serialized form (which include their package ID), and a list of their transitive dependencies.

One of

computation_charge

Type | uint64 Label | optional

Description | The total amount of gas charged for computation during the epoch.

epoch

Type | uint64 Label | optional

Description | The next (to become) epoch ID.

epoch_start_timestamp_ms

Type | uint64 Label | optional

Description | Unix timestamp when epoch started.

non_refundable_storage_fee

Type | uint64 Label | optional

Description | The non-refundable storage fee.

protocol_version

Type | uint64 Label | optional

Description | The protocol version in effect in the new epoch.

storage_charge

Type | uint64 Label | optional

Description | The total amount of gas charged for storage during the epoch.

storage_rebate

Type | uint64 Label | optional

Description | The amount of storage rebate refunded to the txn senders.

ChangedObject

Input/output state of an object that was changed during execution.

Fields

One of

created

Type | Empty

deleted

Type | Empty

exist

Type | ObjectExist

Description | Object existed prior to this transaction.

none

Type | Empty

not_exist

Type | Empty

Description | Object did not exist prior to this transaction.

object_id

Type | ObjectId Label | optional

Description | Required. ID of the object.

object_write

Type | ObjectWrite

Description | Object was written, including all of mutated, created, unwrapped.

package_write

Type | PackageWrite

Description | Package was written.

removed

Type | Empty

Description | Object was removed from the store due to this transaction.

CheckpointCommitment

A commitment made by a checkpoint.

Fields

One of

ecmh_live_object_set

Type | Digest

Description | An elliptic curve multiset hash attesting to the set of objects that comprise the live state of the Sui blockchain.

CheckpointContents

The committed to contents of a checkpoint.

Fields

One of

v1

Type | V1

V1

Version 1 of `CheckpointContents`.

Fields

transactions

Type | CheckpointedTransactionInfo Label | repeated

CheckpointSummary

A header for a checkpoint on the Sui blockchain.

On the Sui network, checkpoints define the history of the blockchain. They are quite similar to the concept of blocks used by other blockchains like Bitcoin or Ethereum. The Sui blockchain, however, forms checkpoints after transaction execution has already happened to provide a certified history of the chain, instead of being formed before execution.

Checkpoints commit to a variety of state, including but not limited to:

- The hash of the previous checkpoint.
- The set of transaction digests, their corresponding effects digests, as well as the set of user signatures that authorized its execution.

- The objects produced by a transaction.
- The set of live objects that make up the current state of the chain.
- On epoch transitions, the next validator committee.

`CheckpointSummary`s themselves don't directly include all of the previous information but they are the top-level type by which all the information is committed to transitively via cryptographic hashes included in the summary. `CheckpointSummary`s are signed and certified by a quorum of the validator committee in a given epoch to allow verification of the chain's state.

Fields

commitments

Type | CheckpointCommitment Label | repeated

Description | Commitments to checkpoint-specific state.

One of

content_digest

Type | Digest Label | optional

Description | The hash of the `CheckpointContents` for this checkpoint.

end_of_epoch_data

Type | EndOfEpochData Label | optional

Description | Extra data only present in the final checkpoint of an epoch.

epoch

Type | uint64 Label | optional

Description | Epoch that this checkpoint belongs to.

epoch_rolling_gas_cost_summary

Type | GasCostSummary Label | optional

Description | The running total gas costs of all transactions included in the current epoch so far until this checkpoint.

previous_digest

Type | Digest Label | optional

Description | The hash of the previous `CheckpointSummary`. This will be `None` only for the first, or genesis, checkpoint.

sequence_number

Type | uint64 Label | optional

Description | The height of this checkpoint.

timestamp_ms

Type | uint64 Label | optional

Description | Timestamp of the checkpoint - number of milliseconds from the Unix epoch Checkpoint timestamps are monotonic, but not strongly monotonic - subsequent checkpoints can have the same timestamp if they originate from the same underlining consensus commit.

total_network_transactions

Type | uint64 Label | optional

Description | Total number of transactions committed since genesis, including those in this checkpoint.

version_specific_data

Type | bytes Label | optional

Description | `CheckpointSummary` is not an evolvable structure - it must be readable by any version of the code. Therefore, to allow extensions to be added to `CheckpointSummary`, opaque data can be added to checkpoints, which can be

deserialized based on the current protocol version.

CheckpointedTransactionInfo

Transaction information committed to in a checkpoint.

Fields

- signatures

  Type | UserSignature Label | repeated

  Description | Set of user signatures that authorized the transaction.

- One of

  - effects

    Type | Digest Label | optional

    Description | Digest of the effects.

  - transaction

    Type | Digest Label | optional

    Description | Digest of the transaction.

CircomG1

A G1 point.

Fields

- One of

  - e0

    Type | Bn254FieldElement Label | optional

    Description | Required.

  - e1

Type | Bn254FieldElement Label | optional

Description | Required.

e2

Type | Bn254FieldElement Label | optional

Description | Required.

CircomG2

A G2 point.

Fields

One of

e00

Type | Bn254FieldElement Label | optional

Description | Required.

e01

Type | Bn254FieldElement Label | optional

Description | Required.

e10

Type | Bn254FieldElement Label | optional

Description | Required.

e11

Type | Bn254FieldElement Label | optional

Description | Required.

e20

Type | Bn254FieldElement Label | optional

Description | Required.

e21

Type | Bn254FieldElement Label | optional

Description | Required.

Command

A single command in a programmable transaction.

Fields

One of

make_move_vector

Type | MakeMoveVector

Description | `forall T: Vec<T> -> vector<T>` Given n-values of the same type, it constructs a vector. For non-objects or an empty vector, the type tag must be specified.

merge_coins

Type | MergeCoins

Description | `(&mut Coin<T>, Vec<Coin<T>>)` It merges n-coins into the first coin.

move_call

Type | MoveCall

Description | A call to either an entry or a public Move function.

publish

Type | Publish

Description | Publishes a Move package. It takes the package bytes and a list of the package's

transitive dependencies to link against on chain.

split_coins

Type | SplitCoins

Description | `(&mut Coin<T>, Vec<u64>)` → `Vec<Coin<T>>` It splits off some amounts into new coins with those amounts.

transfer_objects

Type | TransferObjects

Description | `(Vec<forall T:key+store. T>, address)` It sends n-objects to the specified address. These objects must have store (public transfer) and either the previous owner must be an address or the object must be newly created.

upgrade

Type | Upgrade

Description | Upgrades a Move package. Takes (in order): 1. A vector of serialized modules for the package. 2. A vector of object ids for the transitive dependencies of the new package. 3. The object ID of the package being upgraded. 4. An argument holding the `UpgradeTicket` that must have been produced from an earlier command in the same programmable transaction.

CommandArgumentError

An error with an argument to a command.

Fields

One of

argument

Type | uint32 Label | optional

Description | Required. Position of the problematic argument.

index_out_of_bounds

Type | uint32

Description | Out of bounds access to input or results.

invalid_argument_to_private_entry_function

Type | Empty

Description | Invalid argument to private entry function. Private entry functions cannot take arguments from other Move functions.

invalid_bcs_bytes

Type | Empty

Description | The argument cannot be deserialized into a value of the specified type.

invalid_gas_coin_usage

Type | Empty

Description | Invalid usage of gas coin. The gas coin can only be used by-value with a `TransferObject` command.

invalid_object_by_mut_ref

Type | Empty

Description | Immutable objects cannot be passed by mutable reference, `&mut`.

invalid_object_by_value

Type | Empty

Description | Immutable objects cannot be passed by-value.

invalid_result_arity

Type | uint32

Description | Invalid usage of result. Expected a single result but found either no return value or multiple.

invalid_usage_of_pure_argument

Type | Empty

Description | The argument cannot be instantiated from raw bytes.

invalid_value_usage

Type | Empty

Description | Invalid usage of Move value. - Mutably borrowed values require unique usage. - Immutably borrowed values cannot be taken or borrowed mutably. - Taken values cannot be used again.

secondary_index_out_of_bounds

Type | NestedResult

Description | Out of bounds access to subresult.

shared_object_operation_not_allowed

Type | Empty

Description | Shared object operations such as wrapping, freezing, or converting to owned are not allowed.

type_mismatch

Type | Empty

Description | The type of the value does not match the expected type.

CongestedObjectsError

Set of objects that were congested, leading to the transaction's cancellation.

Fields

congested_objects

Type | ObjectId Label | repeated

Description | Set of congested objects.

ConsensusCommitPrologue

Consensus commit prologue system transaction.

This message can represent V1, V2, and V3 prologue types.

Fields

One of

commit_timestamp_ms

Type | uint64 Label | optional

Description | Unix timestamp from consensus. Present in V1, V2, and V3.

consensus_commit_digest

Type | Digest Label | optional

Description | Digest of consensus output. Present in V2 and V3.

consensus_determined_version_assignments

Type | ConsensusDeterminedVersionAssignments Label | optional

Description | Stores consensus handler determined shared object version assignments. Present in V3.

epoch

Type | uint64 Label | optional

Description | Epoch of the commit prologue transaction. Present in V1, V2, and V3.

round

Type | uint64 Label | optional

Description | Consensus round of the commit. Present in V1, V2, and V3.

sub_dag_index

Type | uint64 Label | optional

Description | The sub DAG index of the consensus commit. This field is populated if there are multiple consensus commits per round. Present in V3.

ConsensusDeterminedVersionAssignments

Version assignments performed by consensus.

Fields

One of

cancelled_transactions

Type | CancelledTransactions

Description | Cancelled transaction version assignment.

Digest

32-byte output of hashing a Sui structure using the Blake2b256 hash function.

Fields

One of

digest

Type | bytes Label | optional

Description | Required. 32-byte hash.

EndOfEpochData

Data, which when included in a `CheckpointSummary`, signals the end of an `Epoch`.

Fields

epoch_commitments

Type | CheckpointCommitment Label | repeated

Description | Commitments to epoch specific state (live object set)

next_epoch_committee

Type | ValidatorCommitteeMember Label | repeated

Description | The set of validators that will be in the `ValidatorCommittee` for the next epoch.

One of

next_epoch_protocol_version

Type | uint64 Label | optional

Description | The protocol version that is in effect during the next epoch.

EndOfEpochTransaction

Set of operations run at the end of the epoch to close out the current epoch and start the next one.

Fields

- **transactions**

  Type | EndOfEpochTransactionKind Label | repeated

EndOfEpochTransactionKind

Operation run at the end of an epoch.

Fields

- **One of**

  - **authenticator_state_create**

    Type | Empty

    Description | Create and initialize the authenticator object used for zklogin.

  - **authenticator_state_expire**

    Type | AuthenticatorStateExpire

    Description | Expire JWKs used for zklogin.

  - **bridge_committee_init**

    Type | uint64

    Description | Initialize the bridge committee.

  - **bridge_state_create**

    Type | Digest

    Description | Create and initialize the bridge object.

  - **change_epoch**

    Type | ChangeEpoch

    Description | End the epoch and start the next one.

deny_list_state_create

Type | Empty

Description | Create and initialize the deny list object.

randomness_state_create

Type | Empty

Description | Create and initialize the randomness object.

Event

An event.

Fields

One of

contents

Type | bytes Label | optional

Description | BCS serialized bytes of the event.

event_type

Type | StructTag Label | optional

Description | The type of the event emitted.

module

Type | Identifier Label | optional

Description | Module name of the top-level function invoked by a `MoveCall` command that triggered this event to be emitted.

package_id

Type | ObjectId Label | optional

Description | Package ID of the top-level function invoked by a `MoveCall` command that triggered this

event to be emitted.

sender

Type | Address Label | optional

Description | Address of the account that sent the transaction where this event was emitted.

ExecutionStatus

The status of an executed transaction.

Fields

One of

status

Type | FailureStatus Label | optional

Description | Optional. The error if `success` is false.

success

Type | bool Label | optional

Description | Required. Indicates if the transaction was successful or not.

FailureStatus

An error that can occur during the execution of a transaction.

Fields

One of

address_denied_for_coin

Type | AddressDeniedForCoinError

Description | Address is denied for this coin type.

arity_mismatch

Type | Empty

Description | Parity mismatch for Move function. The number of arguments does not match the number of parameters.

certificate_denied

Type | Empty

Description | Certificate is on the deny list.

circular_object_ownership

Type | ObjectId

Description | Circular object ownership.

coin_balance_overflow

Type | Empty

Description | Coin balance overflowed an u64.

coin_type_global_pause

Type | string

Description | Coin type is globally paused for use.

command

Type | uint64 Label | optional

Description | The command, if any, during which the error occurred.

command_argument_error

Type | CommandArgumentError

Description | Invalid command argument.

effects_too_large

Type | SizeError

Description | Post-execution errors. Effects from the transaction are too large.

execution_cancelled_due_to_randomness_unavailable

Type | Empty

Description | Certificate is cancelled because randomness could not be generated this epoch.

execution_cancelled_due_to_shared_object_congestion

Type | CongestedObjectsError

Description | Certificate is cancelled due to congestion on shared objects.

feature_not_yet_supported

Type | Empty

Description | Attempted to use feature that is not supported yet.

function_not_found

Type | Empty

Description | Programmable transaction errors. Function not found.

input_object_deleted

Type | Empty

Description | Requested shared object has been deleted.

insufficient_coin_balance

Type | Empty

Description | Coin errors. Insufficient coin balance for requested operation.

insufficient_gas

Type | Empty

Description | Insufficient gas.

invalid_gas_object

Type | Empty

Description | Invalid `Gas` object.

invalid_public_function_return_type

Type | uint32

Description | Invalid public Move function signature. Unsupported return type for return value.

invalid_transfer_object

Type | Empty

Description | Invalid transfer object, object does not have public transfer.

invariant_violation

Type | Empty

Description | Invariant violation.

move_abort

Type | MoveError

Description | Move runtime abort.

move_primitive_runtime_error

Type | MoveError

Description | MoveVm errors. Error from a non-abort instruction. Possible causes: Arithmetic error, stack overflow, max value depth, or similar.

non_entry_function_invoked

Type | Empty

Description | Non-entry function invoked. Move Call must start with an entry function.

object_too_big

Type | SizeError

Description | Move object is larger than the maximum allowed size.

package_too_big

Type | SizeError

Description | Package is larger than the maximum allowed size.

package_upgrade_error

Type | PackageUpgradeError

Description | Invalid package upgrade.

publish_error_non_zero_address

Type | Empty

Description | Publish/Upgrade errors. Publish error, non-zero address. The modules in the package must have their self-addresses set to zero.

publish_upgrade_dependency_downgrade

Type | Empty

Description | Publish or upgrade dependency downgrade. Indirect (transitive) dependency of published or upgraded package has been assigned an on-chain version that is less than the version required by one of the package's transitive dependencies.

publish_upgrade_missing_dependency

Type | Empty

Description | Publish or Upgrade is missing dependency.

shared_object_operation_not_allowed

Type | Empty

Description | The requested shared object operation is not allowed.

sui_move_verification_error

Type | Empty

Description | Sui Move bytecode verification error.

sui_move_verification_timedout

Type | Empty

Description | Sui Move bytecode verification timed out.

type_argument_error

Type | TypeArgumentError

Description | Type argument error.

type_arity_mismatch

Type | Empty

Description | Type parity mismatch for Move function. Mismatch between the number of actual versus expected type arguments.

unused_value_without_drop

Type | NestedResult

Description | Unused result without the drop ability.

vm_invariant_violation

Type | Empty

Description | MoveVm invariant violation.

vm_verification_or_deserialization_error

Type | Empty

Description | Bytecode verification error.

written_objects_too_large

Type | SizeError

Description | Indicates the transaction tried to write objects too large to storage.

GasCostSummary

Summary of gas charges.

Storage is charged independently of computation. There are three parts to the storage charges:

- `storage_cost`: the charge of storage at the time the transaction is executed. The cost of storage is the number of bytes of the objects being mutated multiplied by a variable storage cost per byte.
- `storage_rebate`: the amount a user gets back when manipulating an object. The `storage_rebate` is the `storage_cost` for an object minus fees.
- `non_refundable_storage_fee`: not all the value of the object storage cost is given back to user and there is a small fraction that is kept by the system. This value tracks that charge.

When looking at a gas cost summary the amount charged to the user is `computation_cost + storage_cost - storage_rebate` and that is the amount that is deducted from the gas coins. `non_refundable_storage_fee` is collected from the objects being mutated/deleted and it is tracked by the system in storage funds.

Objects deleted, including the older versions of objects mutated, have the storage field on the objects added up to a pool of "potential rebate". This rebate then is reduced by the "nonrefundable rate" such that: `potential_rebate(storage cost of deleted/mutated objects) = storage_rebate + non_refundable_storage_fee`

Fields

One of

computation_cost

Type | uint64 Label | optional

Description | Cost of computation/execution.

non_refundable_storage_fee

Type | uint64 Label | optional

Description | The fee for the rebate. The portion of the storage rebate kept by the system.

storage_cost

Type | uint64 Label | optional

Description | Storage cost, it's the sum of all storage cost for all objects created or mutated.

storage_rebate

Type | uint64 Label | optional

Description | The amount of storage cost refunded to the user for all objects deleted or mutated in the transaction.

GasPayment

Payment information for executing a transaction.

Fields

objects

Type | ObjectReference Label | repeated

Description | Set of gas objects to use for payment.

One of

budget

Type | uint64 Label | optional

Description | Total budget willing to spend for the execution of a transaction.

owner

Type | Address Label | optional

Description | Owner of the gas objects, either the transaction sender or a sponsor.

price

Type | uint64 Label | optional

Description | Gas unit price to use when charging for computation. Must be greater than or equal to the network's current RGP (reference gas price).

GenesisObject

An object part of the initial chain state.

Fields

One of

object

Type | ObjectData Label | optional

object_id

Type | ObjectId Label | optional

owner

Type | Owner Label | optional

version

Type | uint64 Label | optional

GenesisTransaction

The genesis transaction.

Fields

- **objects**

  Type | GenesisObject Label | repeated

  Description | Set of genesis objects.

I128

A signed 128-bit integer encoded in little-endian using 16-bytes.

Fields

- **One of**

  - **bytes**

    Type | bytes Label | optional

    Description | Required. 16-byte little-endian bytes.

Identifier

A Move identifier.

Identifiers are only valid if they conform to the following ABNF:

```
identifier = (ALPHA *127(ALPHA / DIGIT / UNDERSCORE)) /
             (UNDERSCORE 1*127(ALPHA / DIGIT / UNDERSCORE))
UNDERSCORE = %x95
```

Fields

- **One of**

identifier

Type | string Label | optional

Input

An input to a user transaction.

Fields

One of

immutable_or_owned

Type | ObjectReference

Description | A Move object that is either immutable or address owned.

pure

Type | bytes

Description | A move value serialized as BCS. For normal operations this is required to be a move primitive type and not contain structs or objects.

receiving

Type | ObjectReference

Description | A Move object that is attempted to be received in this transaction.

shared

Type | SharedObjectInput

Description | A Move object whose owner is "Shared".

Jwk

A JSON web key.

Struct that contains info for a JWK. A list of them for different kinds can be retrieved from the JWK endpoint (for example, &#lt;https://www.googleapis.com/oauth2/v3/certs>). The JWK is used to verify the JWT token.

Fields

One of

alg

Type | string Label | optional

Description | Algorithm parameter, https://datatracker.ietf.org/doc/html/rfc7517#section-4.4.

e

Type | string Label | optional

Description | RSA public exponent, https://datatracker.ietf.org/doc/html/rfc7517#section-9.3.

kty

Type | string Label | optional

Description | Key type parameter, https://datatracker.ietf.org/doc/html/rfc7517#section-4.1.

n

Type | string Label | optional

Description | RSA modulus, https://datatracker.ietf.org/doc/html/rfc7517#section-9.3.

JwkId

Key to uniquely identify a JWK.

Fields

One of

iss

Type | string Label | optional

Description | The issuer or identity of the OIDC provider.

kid

Type | string Label | optional

Description | A key ID used to uniquely identify a key from an OIDC provider.

MakeMoveVector

Command to build a Move vector out of a set of individual elements.

Fields

elements

Type | Argument Label | repeated

Description | The set individual elements to build the vector with.

One of

element_type

Type | TypeTag Label | optional

Description | Type of the individual elements. This is required to be set when the type can't be inferred, for example when the set of provided arguments are all pure input values.

MergeCoins

Command to merge multiple coins of the same type into a single coin.

Fields

- coins_to_merge

  Type | Argument Label | repeated

  Description | Set of coins to merge into `coin`. All listed coins must be of the same type and be the same type as `coin`

- One of

  - coin

    Type | Argument Label | optional

    Description | Coin to merge coins into.

ModifiedAtVersion

Indicates that an object was modified at a specific version.

Fields

- One of

  - object_id

    Type | ObjectId Label | optional

    Description | Required. `ObjectId` of the object.

  - version

    Type | uint64 Label | optional

    Description | Required. Version of the object prior to this transaction.

MoveCall

Command to call a Move function.

Functions that can be called by a `MoveCall` command are those that have a function signature that is either `entry` or `public` (which don't

have a reference return type).

Fields

- **arguments**

  Type | Argument Label | repeated

  Description | The arguments to the function.

- **type_arguments**

  Type | TypeTag Label | repeated

  Description | The type arguments to the function.

- **One of**

  - **function**

    Type | Identifier Label | optional

    Description | The function to be called.

  - **module**

    Type | Identifier Label | optional

    Description | The specific module in the package containing the function.

  - **package**

    Type | ObjectId Label | optional

    Description | The package containing the module and function.

MoveError

Error that occurred in Move.

Fields

- **One of**

abort_code

Type | uint64 Label | optional

Description | Abort code from Move.

location

Type | MoveLocation Label | optional

Description | Location in Move where the error occurred.

MoveField

Fields

One of

name

Type | Identifier Label | optional

value

Type | MoveValue Label | optional

MoveLocation

Location in Move bytecode where an error occurred.s

Fields

One of

function

Type | uint32 Label | optional

Description | Required. The function index.

function_name

Type | Identifier Label | optional

Description | Optional. The name of the function, if available.

instruction

Type | uint32 Label | optional

Description | Required. Offset of the instruction where the error occurred.

module

Type | Identifier Label | optional

Description | Required. The module name.

package

Type | ObjectId Label | optional

Description | Required. The package ID.

MoveModule

Module defined by a package.

Fields

One of

contents

Type | bytes Label | optional

Description | Serialized bytecode of the module.

name

Type | Identifier Label | optional

Description | Name of the module.

MovePackage

A Move package.

Fields

linkage_table

Type | UpgradeInfo Label | repeated

Description | For each dependency, maps original package ID to the info about the (upgraded) dependency version that this package is using.

modules

Type | MoveModule Label | repeated

Description | Set of modules defined by this package.

type_origin_table

Type | TypeOrigin Label | repeated

Description | Maps struct/module to a package version where it was first defined, stored as a vector for simple serialization and deserialization.

One of

id

Type | ObjectId Label | optional

Description | Address or ID of this package.

version

Type | uint64 Label | optional

Description | Version of the package.

MoveStruct

A Move struct.

Fields

One of

contents

Type | bytes Label | optional

Description | BCS bytes of a Move struct value.

has_public_transfer

Type | bool Label | optional

Description | DEPRECATED this field is no longer used to determine whether a tx can transfer this object. Instead, it is always calculated from the objects type when loaded in execution.

object_id

Type | ObjectId Label | optional

Description | `ObjectId` for this object.

object_type

Type | StructTag Label | optional

Description | The type of this object.

version

Type | uint64 Label | optional

Description | Version of the object.

MoveStructValue

Fields

fields

Type | MoveField Label | repeated

One of

struct_type

Type | StructTag Label | optional

MoveValue

Fields

One of

- address

  Type | Address

- bool

  Type | bool

- signer

  Type | Address

- struct

  Type | MoveStructValue

- u128

  Type | U128

- u16

  Type | uint32

- u256

  Type | U256

- u32

  Type | uint32

- u64

  Type | uint64

u8

Type | uint32

variant

Type | MoveVariant

vector

Type | MoveVector

MoveVariant

Fields

fields

Type | MoveField Label | repeated

One of

enum_type

Type | StructTag Label | optional

tag

Type | uint32 Label | optional

variant_name

Type | Identifier Label | optional

MoveVector

Fields

values

Type | MoveValue Label | repeated

MultisigAggregatedSignature

Aggregated signature from members of a multisig committee.

Fields

- **signatures**

  Type | MultisigMemberSignature Label | repeated

  Description | The plain signatures encoded with signature scheme. The signatures must be in the same order as they are listed in the committee.

- **One of**

  - **bitmap**

    Type | uint32 Label | optional

    Description | Required. Bitmap indicating which committee members contributed to the signature.

  - **committee**

    Type | MultisigCommittee Label | optional

    Description | Required. The committee to use to validate this signature.

  - **legacy_bitmap**

    Type | RoaringBitmap Label | optional

    Description | Optional. If present, means this signature's on-chain format uses the old legacy multisig format.

MultisigCommittee

A multisig committee.

Fields

members

Type | MultisigMember Label | repeated

Description | A list of committee members and their corresponding weight.

One of

threshold

Type | uint32 Label | optional

Description | Required. The threshold of signatures needed to validate a signature from this committee.

MultisigMember

A member in a multisig committee.

Fields

One of

public_key

Type | MultisigMemberPublicKey Label | optional

Description | Required. The public key of the committee member.

weight

Type | uint32 Label | optional

Description | Required. The weight of this member's signature.

MultisigMemberPublicKey

Set of valid public keys for multisig committee members.

Fields

One of

ed25519

Type | bytes

Description | An ed25519 public key

secp256k1

Type | bytes

Description | A secp256k1 public key

secp256r1

Type | bytes

Description | A secp256r1 public key

zklogin

Type | ZkLoginPublicIdentifier

Description | A zklogin public identifier

MultisigMemberSignature

A signature from a member of a multisig committee.

Fields

One of

ed25519

Type | bytes

Description | An ed25519 signature.

secp256k1

Type | bytes

Description | A secp256k1 signature.

secp256r1

Type | bytes

Description | A secp256r1 signature.

zklogin

Type | ZkLoginAuthenticator

Description | A zklogin signature.

NestedResult

An argument type for a nested result.

Fields

One of

result

Type | uint32 Label | optional

Description | The command index.

subresult

Type | uint32 Label | optional

Description | The index into the command's output.

Object

An object on the Sui blockchain.

Fields

One of

object

Type | ObjectData Label | optional

object_id

Type | ObjectId Label | optional

Description | `ObjectId` for this object.

owner

Type | Owner Label | optional

Description | Owner of the object.

previous_transaction

Type | Digest Label | optional

Description | The digest of the transaction that created or last mutated this object

storage_rebate

Type | uint64 Label | optional

Description | The amount of SUI to rebate if this object gets deleted. This number is re-calculated each time the object is mutated based on the present storage gas price.

version

Type | uint64 Label | optional

Description | Version of the object.

ObjectData

Object data, either a package or struct.

Fields

One of

package

Type | MovePackage

struct

Type | MoveStruct

ObjectExist

Information about the old version of the object.

Fields

One of

digest

Type | Digest Label | optional

Description | Required. Digest of the object.

owner

Type | Owner Label | optional

Description | Required. Owner of the object.

version

Type | uint64 Label | optional

Description | Required. Version of the object.

ObjectId

Unique identifier for an object on the Sui blockchain.

An `ObjectId` is a 32-byte identifier used to uniquely identify an object on the Sui blockchain.

Fields

One of

object_id

Type | bytes Label | optional

Description | Required. 32-byte object-id.

ObjectReference

Reference to an object.

Fields

One of

digest

Type | Digest Label | optional

Description | The digest of this object.

object_id

Type | ObjectId Label | optional

Description | The object ID of this object.

version

Type | uint64 Label | optional

Description | The version of this object.

ObjectReferenceWithOwner

An object reference with owner information.

Fields

One of

owner

Type | Owner Label | optional

Description | Required. `Owner`.

reference

Type | ObjectReference Label | optional

Description | Required. `ObjectReference`.

ObjectWrite

Object write, including all of mutated, created, unwrapped.

Fields

One of

digest

Type | Digest Label | optional

Description | Required. Digest of the new version of the object.

owner

Type | Owner Label | optional

Description | Required. Owner of the new version of the object.

Owner

Enum of different types of ownership for an object.

Fields

One of

address

Type | Address

Description | Object is exclusively owned by a single address, and is mutable.

immutable

Type | Empty

Description | Object is immutable, and hence ownership doesn't matter.

object

Type | ObjectId

Description | Object is exclusively owned by a single object, and is mutable.

shared

Type | uint64

Description | Object is shared, can be used by any address, and is mutable.

PackageIdDoesNotMatch

Package ID does not match `PackageId` in upgrade ticket.

Fields

One of

package_id

Type | ObjectId Label | optional

Description | Required. The package ID.

ticket_id

Type | ObjectId Label | optional

Description | Required. The ticket ID.

PackageUpgradeError

An error with a upgrading a package.

Fields

One of

digets_does_not_match

Type | Digest

Description | Digest in upgrade ticket and computed digest differ.

incompatible_upgrade

Type | Empty

Description | Package upgrade is incompatible with previous version.

not_a_package

Type | ObjectId

Description | Object is not a package.

package_id_does_not_match

Type | PackageIdDoesNotMatch

Description | Package ID does not match `PackageId` in upgrade ticket.

unable_to_fetch_package

Type | ObjectId

Description | Unable to fetch package.

unknown_upgrade_policy

Type | uint32

Description | Upgrade policy is not valid.

PackageWrite

Package write.

Fields

One of

digest

Type | Digest Label | optional

Description | Required. Digest of the new package.

version

Type | uint64 Label | optional

Description | Version of the new package.

PasskeyAuthenticator

A passkey authenticator.

See struct.PasskeyAuthenticator for more information on the requirements on the shape of the `client_data_json` field.

Fields

One of

authenticator_data

Type | bytes Label | optional

Description | Required. Opaque authenticator data for this passkey signature. See Authenticator Data for more information on this field.

client_data_json

Type | string Label | optional

Description | Required. Structured, unparsed, JSON for this passkey signature. See CollectedClientData for more information on this field.

signature

Type | SimpleSignature Label | optional

Description | Required. A secp256r1 signature.

ProgrammableTransaction

A user transaction.

Contains a series of native commands and Move calls where the results of one command can be used in future commands.

Fields

commands

Type | Command Label | repeated

Description | The commands to be executed sequentially. A failure in any command results in the failure of the entire transaction.

inputs

Type | Input Label | repeated

Description | Input objects or primitive values.

Publish

Command to publish a new Move package.

Fields

dependencies

Type | ObjectId Label | repeated

Description | Set of packages that the to-be published package depends on.

modules

Type | bytes Label | repeated

Description | The serialized Move modules.

RandomnessStateUpdate

Randomness update.

Fields

One of

epoch

Type | uint64 Label | optional

Description | Epoch of the randomness state update transaction.

random_bytes

Type | bytes Label | optional

Description | Updated random bytes.

randomness_object_initial_shared_version

Type | uint64 Label | optional

Description | The initial version of the randomness object that it was shared at.

randomness_round

Type | uint64 Label | optional

Description | Randomness round of the update.

ReadOnlyRoot

Read-only shared object from the input.

Fields

One of

digest

Type | Digest Label | optional

Description | Required. Digest of the shared object.

version

Type | uint64 Label | optional

Description | Required. Version of the shared object.

RoaringBitmap

A RoaringBitmap. See RoaringFormatSpec for the specification for the serialized format of `RoaringBitmap`s.

Fields

One of

bitmap

Type | bytes Label | optional

Description | Required. Serialized `RoaringBitmap`.

SharedObjectInput

A shared object input.

Fields

One of

initial_shared_version

Type | uint64 Label | optional

Description | Initial version of the object when it was shared.

mutable

Type | bool Label | optional

Description | Controls whether the caller asks for a mutable reference to the shared object.

object_id

Type | ObjectId Label | optional

Description | `ObjectId` of the shared object.

SimpleSignature

A basic signature.

Can either be an ed25519, secp256k1, or secp256r1 signature with corresponding public key.

Fields

One of

public_key

Type | bytes Label | optional

Description | Required. Public key bytes.

scheme

Type | SignatureScheme Label | optional

Description | Required. Signature scheme of the signature and public key.

signature

Type | bytes Label | optional

Description | Required. Signature bytes.

SizeError

A size error.

Fields

One of

max_size

Type | uint64 Label | optional

Description | Required. The maximum allowable size.

size

Type | uint64 Label | optional

Description | Required. The offending size.

SplitCoins

Command to split a single coin object into multiple coins.

Fields

amounts

Type | Argument Label | repeated

Description | The amounts to split off.

One of

coin

Type | Argument Label | optional

Description | The coin to split.

StructTag

Type information for a Move struct.

Fields

type_parameters

Type | TypeTag Label | repeated

Description | List of type parameters, if any.

One of

address

Type | Address Label | optional

Description | Address of the package where this type was defined.

module

Type | Identifier Label | optional

Description | Name of the module where this type was defined.

name

Type | Identifier Label | optional

Description | Name of the type itself.

SystemPackage

System package.

Fields

dependencies

Type | ObjectId Label | repeated

Description | Package dependencies.

modules

Type | bytes Label | repeated

Description | Move modules.

One of

version

Type | uint64 Label | optional

Description | Version of the package.

Transaction

A transaction.

Fields

- One of
  - v1

    Type | TransactionV1

TransactionV1

Version 1 of `Transaction`.

Fields

- One of
  - expiration

    Type | TransactionExpiration Label | optional
  - gas_payment

    Type | GasPayment Label | optional
  - kind

    Type | TransactionKind Label | optional
  - sender

    Type | Address Label | optional

TransactionEffects

The output or effects of executing a transaction.

Fields

One of

v1

Type | TransactionEffectsV1

v2

Type | TransactionEffectsV2

TransactionEffectsV1

Version 1 of `TransactionEffects`.

Fields

created

Type | ObjectReferenceWithOwner Label | repeated

Description | `ObjectReference` and owner of new objects created.

deleted

Type | ObjectReference Label | repeated

Description | Object refs of objects now deleted (the new refs).

dependencies

Type | Digest Label | repeated

Description | The set of transaction digests this transaction depends on.

modified_at_versions

Type | ModifiedAtVersion Label | repeated

Description | The version that every modified (mutated or deleted) object had before it was modified by this transaction.

mutated

Type | ObjectReferenceWithOwner Label | repeated

Description | `ObjectReference` and owner of mutated objects, including gas object.

shared_objects

Type | ObjectReference Label | repeated

Description | The object references of the shared objects used in this transaction. Empty if no shared objects were used.

unwrapped

Type | ObjectReferenceWithOwner Label | repeated

Description | `ObjectReference` and owner of objects that are unwrapped in this transaction. Unwrapped objects are objects that were wrapped into other objects in the past, and just got extracted out.

unwrapped_then_deleted

Type | ObjectReference Label | repeated

Description | Object refs of objects previously wrapped in other objects but now deleted.

wrapped

Type | ObjectReference Label | repeated

Description | Object refs of objects now wrapped in other objects.

One of

epoch

Type | uint64 Label | optional

Description | The epoch when this transaction was executed.

events_digest

Type | Digest Label | optional

Description | The digest of the events emitted during execution, can be `None` if the transaction

does not emit any event.

gas_object

Type | ObjectReferenceWithOwner Label | optional

Description | The updated gas object reference. Have a dedicated field for convenient access. It's also included in mutated.

gas_used

Type | GasCostSummary Label | optional

Description | The gas used by this transaction.

status

Type | ExecutionStatus Label | optional

Description | The status of the execution.

transaction_digest

Type | Digest Label | optional

Description | The transaction digest.

TransactionEffectsV2

Version 2 of `TransactionEffects`.

Fields

changed_objects

Type | ChangedObject Label | repeated

Description | Objects whose state are changed in the object store.

dependencies

Type | Digest Label | repeated

Description | The set of transaction digests this transaction depends on.

unchanged_shared_objects

Type | UnchangedSharedObject Label | repeated

Description | Shared objects that are not mutated in this transaction. Unlike owned objects, read-only shared objects' version are not committed in the transaction, and in order for a node to catch up and execute it without consensus sequencing, the version needs to be committed in the effects.

One of

auxiliary_data_digest

Type | Digest Label | optional

Description | Auxiliary data that are not protocol-critical, generated as part of the effects but are stored separately. Storing it separately allows us to avoid bloating the effects with data that are not critical. It also provides more flexibility on the format and type of the data.

epoch

Type | uint64 Label | optional

Description | The epoch when this transaction was executed.

events_digest

Type | Digest Label | optional

Description | The digest of the events emitted during execution, can be `None` if the transaction does not emit any event.

gas_object_index

Type | uint32 Label | optional

Description | The updated gas object reference, as an index into the `changed_objects` vector. Having

a dedicated field for convenient access. System transaction that don't require gas will leave this as `None`.

gas_used

Type | GasCostSummary Label | optional

Description | The gas used by this transaction.

lamport_version

Type | uint64 Label | optional

Description | The version number of all the written Move objects by this transaction.

status

Type | ExecutionStatus Label | optional

Description | The status of the execution.

transaction_digest

Type | Digest Label | optional

Description | The transaction digest.

TransactionEvents

Events emitted during the successful execution of a transaction.

Fields

events

Type | Event Label | repeated

TransactionExpiration

A TTL for a transaction.

Fields

One of

epoch

Type | uint64

Description | Validators won't sign and execute transaction unless the expiration epoch is greater than or equal to the current epoch.

none

Type | Empty

Description | The transaction has no expiration.

TransactionKind

Transaction type.

Fields

One of

authenticator_state_update

Type | AuthenticatorStateUpdate

Description | Update set of valid JWKs used for zklogin.

change_epoch

Type | ChangeEpoch

Description | System transaction used to end an epoch. The `ChangeEpoch` variant is now deprecated (but the `ChangeEpoch` struct is still used by `EndOfEpochTransaction`).

consensus_commit_prologue_v1

Type | ConsensusCommitPrologue

Description | V1 consensus commit update.

consensus_commit_prologue_v2

Type | ConsensusCommitPrologue

Description | V2 consensus commit update.

consensus_commit_prologue_v3

Type | ConsensusCommitPrologue

Description | V3 consensus commit update.

end_of_epoch

Type | EndOfEpochTransaction

Description | Set of operations to run at the end of the epoch to close out the current epoch and start the next one.

genesis

Type | GenesisTransaction

Description | Transaction used to initialize the chain state. Only valid if in the genesis checkpoint (0) and if this is the very first transaction ever executed on the chain.

programmable_transaction

Type | ProgrammableTransaction

Description | A user transaction comprised of a list of native commands and Move calls.

randomness_state_update

Type | RandomnessStateUpdate

Description | Randomness update.

TransferObjects

Command to transfer ownership of a set of objects to an address.

Fields

- **objects**

  Type | Argument Label | repeated

  Description | Set of objects to transfer.

- **One of**

  - **address**

    Type | Argument Label | optional

    Description | The address to transfer ownership to.

TypeArgumentError

Type argument error.

Fields

- **One of**

  - **constraint_not_satisfied**

    Type | Empty

    Description | A type provided did not match the specified constraint.

  - **type_argument**

    Type | uint32 Label | optional

    Description | Required. Index of the problematic type argument.

  - **type_not_found**

    Type | Empty

    Description | A type was not found in the module specified.

TypeOrigin

Identifies a struct and the module it was defined in.

Fields

One of

module_name

Type | Identifier Label | optional

package_id

Type | ObjectId Label | optional

struct_name

Type | Identifier Label | optional

TypeTag

Type of a Move value.

Fields

One of

address

Type | Empty

bool

Type | Empty

signer

Type | Empty

struct

Type | StructTag

u128

Type | Empty

u16

Type | Empty

u256

Type | Empty

u32

Type | Empty

u64

Type | Empty

u8

Type | Empty

vector

Type | TypeTag

U128

An unsigned 128-bit integer encoded in little-endian using 16-bytes.

Fields

One of

bytes

Type | bytes Label | optional

Description | Required. 16-byte little-endian bytes.

U256

An unsigned 256-bit integer encoded in little-endian using 32-bytes.

Fields

One of

bytes

Type | bytes Label | optional

Description | Required. 16-byte little-endian bytes.

UnchangedSharedObject

A shared object that wasn't changed during execution.

Fields

One of

cancelled

Type | uint64

Description | Shared objects that was congested and resulted in this transaction being cancelled.

mutate_deleted

Type | uint64

Description | Deleted shared objects that appear mutably/owned in the input.

object_id

Type | ObjectId Label | optional

Description | Required. ObjectId of the shared object.

per_epoch_config

Type | Empty

Description | Read of a per-epoch config object that should remain the same during an epoch.

read_deleted

Type | uint64

Description | Deleted shared objects that appear as read-only in the input.

read_only_root

Type | ReadOnlyRoot

Description | Read-only shared object from the input.

Upgrade

Command to upgrade an already published package.

Fields

dependencies

Type | ObjectId Label | repeated

Description | Set of packages that the to-be published package depends on.

modules

Type | bytes Label | repeated

Description | The serialized Move modules.

One of

package

Type | ObjectId Label | optional

Description | Package ID of the package to upgrade.

ticket

Type | Argument Label | optional

Description | Ticket authorizing the upgrade.

UpgradeInfo

Upgraded package info for the linkage table.

Fields

One of

original_id

Type | ObjectId Label | optional

Description | ID of the original package.

upgraded_id

Type | ObjectId Label | optional

Description | ID of the upgraded package.

upgraded_version

Type | uint64 Label | optional

Description | Version of the upgraded package.

UserSignature

A signature from a user.

Fields

One of

multisig

Type | MultisigAggregatedSignature

passkey

Type | PasskeyAuthenticator

simple

Type | SimpleSignature

zklogin

Type | ZkLoginAuthenticator

ValidatorAggregatedSignature

An aggregated signature from multiple validators.

Fields

One of

bitmap

Type | RoaringBitmap Label | optional

Description | Required. Bitmap indicating which members of the committee contributed to this signature.

epoch

Type | uint64 Label | optional

Description | Required. The epoch when this signature was produced. This can be used to lookup the `ValidatorCommittee` from this epoch to verify this signature.

signature

Type | bytes Label | optional

Description | Required. The 48-byte Bls12381 aggregated signature.

ValidatorCommittee

The validator set for a particular epoch.

Fields

members

Type | ValidatorCommitteeMember Label | repeated

Description | The committee members.

One of

epoch

Type | uint64 Label | optional

Description | Required. The epoch where this committee governs.

ValidatorCommitteeMember

A member of a validator committee.

Fields

One of

public_key

Type | bytes Label | optional

Description | Required. The 96-byte Bls12381 public key for this validator.

stake

Type | uint64 Label | optional

Description | Required. Stake weight this validator possesses.

VersionAssignment

Object version assignment from consensus.

Fields

One of

object_id

Type | ObjectId Label | optional

Description | `ObjectId` of the object.

version

Type | uint64 Label | optional

Description | Assigned version.

ZkLoginAuthenticator

A zklogin authenticator.

Fields

One of

inputs

Type | ZkLoginInputs Label | optional

Description | Required. Zklogin proof and inputs required to perform proof verification.

max_epoch

Type | uint64 Label | optional

Description | Required. Maximum epoch for which the proof is valid.

signature

Type | SimpleSignature Label | optional

Description | Required. User signature with the public key attested to by the provided proof.

ZkLoginClaim

A claim of the iss in a zklogin proof.

Fields

One of

index_mod_4

Type | uint32 Label | optional

Description | Required.

value

Type | string Label | optional

Description | Required.

ZkLoginInputs

A zklogin groth16 proof and the required inputs to perform proof verification.

Fields

One of

address_seed

Type | Bn254FieldElement Label | optional

Description | Required.

header_base64

Type | string Label | optional

Description | Required.

iss_base64_details

Type | ZkLoginClaim Label | optional

Description | Required.

proof_points

Type | ZkLoginProof Label | optional

Description | Required.

ZkLoginProof

A zklogin groth16 proof.

Fields

One of

a

Type | CircomG1 Label | optional

Description | Required.

b

Type | CircomG2 Label | optional

Description | Required.

c

Type | CircomG1 Label | optional

Description | Required.

ZkLoginPublicIdentifier

Public key equivalent for zklogin authenticators.

Fields

One of

address_seed

Type | Bn254FieldElement Label | optional

Description | Required.

iss

Type | string Label | optional

Description | Required.

google/protobuf/empty.proto

Empty

A generic empty message that you can re-use to avoid defining duplicated empty messages in your APIs. A typical example is to use it as the request or the response type of an API method. For instance:

```
service Foo {
  rpc Bar(google.protobuf.Empty) returns
(google.protobuf.Empty);
}
```

google/protobuf/timestamp.proto

Timestamp

A Timestamp represents a point in time independent of any time zone or calendar, represented as seconds and fractions of seconds at nanosecond resolution in UTC Epoch time. It is encoded using the Proleptic Gregorian Calendar which extends the Gregorian calendar backwards to year one. It is encoded assuming all minutes are 60 seconds long, i.e. leap seconds are "smeared" so that no leap second table is needed for interpretation. Range is from `0001-01-01T00:00:00Z` to `9999-12-31T23:59:59.999999999Z`. Restricting to that range ensures that conversion to and from RFC 3339 date strings is possible. See https://www.ietf.org/rfc/rfc3339.txt.

Examples

Example 1: Compute Timestamp from POSIX `time()`.

```
Timestamp timestamp;
timestamp.set_seconds(time(NULL));
timestamp.set_nanos(0);
```

Example 2: Compute Timestamp from POSIX `gettimeofday()`.

```
struct timeval tv;
gettimeofday(&tv, NULL);

Timestamp timestamp;
timestamp.set_seconds(tv.tv_sec);
timestamp.set_nanos(tv.tv_usec * 1000);
```

Example 3: Compute Timestamp from Win32 `GetSystemTimeAsFileTime()`.

```
FILETIME ft;
GetSystemTimeAsFileTime(&ft);
UINT64 ticks = (((UINT64)ft.dwHighDateTime) &#lt;&#lt; 32) |
ft.dwLowDateTime;

// A Windows tick is 100 nanoseconds. Windows epoch 1601-01-
01T00:00:00Z
// is 11644473600 seconds before Unix epoch 1970-01-
01T00:00:00Z.
Timestamp timestamp;
timestamp.set_seconds((INT64) ((ticks / 10000000) -
11644473600LL));
timestamp.set_nanos((INT32) ((ticks % 10000000) * 100)); //
```

Example 4: Compute Timestamp from Java `System.currentTimeMillis()`.

```
long millis = System.currentTimeMillis();

Timestamp timestamp = Timestamp.newBuilder().setSeconds(millis
/ 1000)
    .setNanos((int) ((millis % 1000) * 1000000)).build();
```

Example 5: Compute Timestamp from current time in Python.

```
timestamp = Timestamp()
timestamp.GetCurrentTime()
```

JSON Mapping

In JSON format, the `Timestamp` type is encoded as a string in the RFC 3339 format. That is, the format is `{year}-{month}-{day}T{hour}:{min}:{sec}[.{frac_sec}]Z` where `{year}` is always expressed using four digits while `{month}`, `{day}`, `{hour}`, `{min}`, and `{sec}` are zero-padded to two digits each. The fractional seconds, which can go up to 9 digits (so up to 1 nanosecond resolution), are optional. The "Z" suffix indicates the timezone ("UTC"); the timezone is required, though only UTC (as indicated by "Z") is presently supported.

For example, `2017-01-15T01:30:15.01Z` encodes 15.01 seconds past 01:30 UTC on January 15, 2017.

In JavaScript, you can convert a `Date` object to this format using the standard toISOString() method. In Python, you can convert a standard `datetime.datetime` object to this format using `strftime` with the time format spec `%Y-%m-%dT%H:%M:%S.%fZ`. Likewise, in Java, you can use the Joda Time's `ISODateTimeFormat.dateTime()` to obtain a formatter capable of generating timestamps in this format.

Fields

nanos

Type | int32

Description | Non-negative fractions of a second at nanosecond resolution. Negative second values with fractions must still have non-negative nano values that count forward in time. Must be from 0 to 999,999,999 inclusive.

seconds

Type | int64

Description | Represents seconds of UTC time since Unix epoch `1970-01-01T00:00:00Z`. Must be from `0001-01-01T00:00:00Z` to `9999-12-31T23:59:59Z` inclusive.

Scalar Value Types

double

C++	C#	Go	Java	PHP
double	double	float64	double	float

Python	Ruby
float	Float

float

C++	C#	Go	Java	PHP
float	float	float32	float	float

Python	Ruby
float	Float

int32

Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint32 instead.

C++	C#	Go	Java	PHP
int32	int	int32	int	integer

Python	Ruby
int	Bignum or Fixnum (as required)

int64

Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint64 instead.

C++	C#	Go	Java	PHP
int64	long	int64	long	integer/string

Python	Ruby
int/long	Bignum

uint32

Uses variable-length encoding.

C++	C#	Go	Java	PHP
uint32	uint	uint32	int	integer

Python	Ruby
int/long	Bignum or Fixnum (as required)

uint64

Uses variable-length encoding.

C++	C#	Go	Java	PHP
uint64	ulong	uint64	long	integer/string

Python	Ruby
int/long	Bignum or Fixnum (as required)

sint32

Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int32s.

C++	C#	Go	Java	PHP
int32	int	int32	int	integer

Python	Ruby
int	Bignum or Fixnum (as required)

sint64

Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int64s.

C++	C#	Go	Java	PHP
int64	long	int64	long	integer/string

Python	Ruby
int/long	Bignum

fixed32

Always four bytes. More efficient than uint32 if values are often greater than 2^28.

C++	C#	Go	Java	PHP
uint32	uint	uint32	int	integer

Python	Ruby

int	Bignum or Fixnum (as required)

fixed64

Always eight bytes. More efficient than uint64 if values are often greater than 2^56.

C++	C#	Go	Java	PHP
uint64	ulong	uint64	long	integer/string

Python	Ruby
int/long	Bignum

sfixed32

Always four bytes.

C++	C#	Go	Java	PHP
int32	int	int32	int	integer

Python	Ruby
int	Bignum or Fixnum (as required)

sfixed64

Always eight bytes.

C++	C#	Go	Java	PHP
int64	long	int64	long	integer/string

Python	Ruby
int/long	Bignum

bool

C++	C#	Go	Java	PHP
bool	bool	bool	boolean	boolean

Python	Ruby
boolean	TrueClass/FalseClass

string

A string must always contain UTF-8 encoded or 7-bit ASCII text.

C++	C#	Go	Java	PHP
string	string	string	String	string

Python	Ruby
str/unicode	String (UTF-8)

bytes

May contain any arbitrary sequence of bytes.

C++	C#	Go	Java	PHP
string	ByteString	[]byte	ByteString	string

Python	Ruby

str	String (ASCII-8BIT)

Edit this page

© 2025 SUI FOUNDATION | DOCUMENTATION DISTRIBUTED UNDER CC BY 4.0