

Interchainquery

[Suggest Edits](#)

Interchain Query

Abstract

Stride uses interchain queries and interchain accounts to perform multichain liquid staking. The interchainquery module creates a framework that allows other modules to query other appchains using IBC. The interchainquery module is used to make bank balance ICQ queries to withdrawal account every N. The callback triggers ICA bank sends for 90% of the rewards to the delegation account and 10% to the stride hostzone revenue account. The ICA bank send logic is in `x/stakeibc/keeper/callbacks.go`.

Contents

1. [Concepts](#)
2. [State](#)
3. [Events](#)
4. [Keeper](#)
5. [Msgs](#)
6. [Queries](#)

State

The interchainquery module keeps Query objects and modifies the information from query to query, as defined in `proto/interchainquery/v1/genesis.proto`

InterchainQuery information type

Query has information types that pertain to the query itself. Query keeps the following:

1. id
2. : query identification string.
3. connection_id
4. : id of the connection between the controller and host chain.
5. chain_id
6. : id of the queried chain.
7. query_type
8. : type of interchain query (e.g. bank store query)
9. request_data
10. : serialized request information (e.g. the address with which to query)
11. callback_module
12. : name of the module that will handle the callback
13. callback_id
14. : ID for the function that will be called after the response is returned
15. callback_data
16. : optional serialized data associated with the callback
17. timeout_policy
18. : specifies how to handle a timeout (fail the query, retry the query, or execute the callback with a timeout)
19. timeout_duration
20. : the relative time from the current block with which the query should timeout
21. timeout_timestamp
22. : the absolute time at which the query times out
23. request_sent
24. : boolean indicating whether the query event has been emitted (and can be identified by a relayer)
25. submission_height
26. : the light client hight of the queried chain at the time of query submission

DataPoint has information types that pertain to the data that is queried. DataPoint keeps the following:

1. id
2. keeps the identification string of the datapoint
3. remote_height
4. keeps the block height of the queried chain

5. local_height
6. keeps the block height of the querying chain
7. value
8. keeps the bytecode value of the data retrieved by the Query

Events

The interchainquery module emits an event at the end of every stride_epoch s (e.g. 15 minutes on local testnet).

The purpose of this event is to send interchainqueries that query data about staking rewards, which Stride uses to reinvest (aka autocompound) staking rewards.

```
Go event := sdk . NewEvent (sdk . EventTypeMessage ,sdk . NewAttribute (sdk . AttributeKeyModule ,types .
AttributeValueCategory ),sdk . NewAttribute (sdk . AttributeKeyAction ,types . AttributeValueQuery ),sdk . NewAttribute
(types . AttributeKeyQueryId ,queryInfo . Id ),sdk . NewAttribute (types . AttributeKeyChainId ,queryInfo . ChainId ),sdk .
NewAttribute (types . AttributeKeyConnectionId ,queryInfo . ConnectionId ),sdk . NewAttribute (types . AttributeKeyType
,queryInfo . QueryType ),sdk . NewAttribute (types . AttributeKeyHeight ,"0" ),sdk . NewAttribute (types .
AttributeKeyRequest ,hex . EncodeToString (queryInfo . Request )), )
```

Keeper

Keeper Functions

interchainquery/keeper/ module provides utility functions to manage ICQs

```
Go // GetQuery returns query GetQuery (ctx sdk . Context ,id string ) (types . Query ,bool )// SetQuery set query info
SetQuery (ctx sdk . Context ,query types . Query )// DeleteQuery delete query info DeleteQuery (ctx sdk . Context ,id string
)// IterateQueries iterate through queries IterateQueries (ctx sdk . Context ,fn func (index int64 ,queryInfo types . Query )
(stop bool ))// AllQueries returns every queryInfo in the store AllQueries (ctx sdk . Context ) []types . Query
```

Msgs

protobuf // SubmitQueryResponse is used to return the query response back to Stride message MsgSubmitQueryResponse
{ string chain_id = 1; string query_id = 2; bytes result = 3; tendermint.crypto.ProofOps proof_ops = 4; int64 height = 5; string
from_address = 6; }

Queries

protobuf // Query PendingQueries lists all queries that have been requested (i.e. emitted) // but have not had a response
submitted yet message QueryPendingQueriesRequest {} Updated 7 months ago

[Icacallbacks Epochs](#) Did this page help you? Yes No * [Table of Contents](#) * * [Interchain Query](#) * * * [Abstract](#) * * * [Contents](#) * *
* [State](#) * * * [Events](#) * * * [Keeper](#) * * * [Msgs](#) * * * [Queries](#)