Cosmos IBC Integration

note This section presented an overview of how IBC can be used to make an oracle data request on BandChain.

For more information on IBC itself, its architecture, and other related topics, please see Cosmos' Interchain Standardsdocumentation .

IBC Overview

In addition to our ownlite client protocol , we also allow interaction with our data oracle through Cosmos'Inter-Blockchain-Communication , or IBC, protocol. This protocol allows other IBC-compatible blockchains to request data from BandChain.

BandChain-Specific IBC Data Packet

OracleRequestPacketData

OracleRequestPacketData is a data packet sent by a blockchain to BandChain's oracle to request data. It contains the following parameters:

Parameter Type Description ClientID string The unique identifier of this oracle request, as specified by the client. This same unique ID will be sent back to the requester with the oracle response OracleScriptID int64 The unique identifier number assigned to the oracle script when it was first registered on Bandchain Calldata string The data that was passed over to the oracle script to use during its execution eg. list of requested symbols and multiplier AskCount uint64 The number of validators that are requested to respond to this request MinCount uint64 The minimum number of validators necessary for the request to proceed to the execution phase FeeLimit sdk.Coins The maximum tokens that will be paid to all data source providers PrepareGas uint64 The amount of gas to pay to prepare raw requests ExecuteGas uint64 The amount of gas reserved for executing the oracle script during execution phase

OracleResponsePacketData

Subsequently, this is the packet that will be relayed from BandChain back to the requester's chain. It contains information on the response parameters as well as the requested data itself.

Parameter Type Description ClientID string The unique identifier of this oracle request, as specified by the client. This matches the ID stated in the correspondingOracleRequestPacketData RequestID int64 The unique identifier number of the particular request AnsCount uint64 The number of validators that answers the request, retrieved the data, and submitted a report RequestTime int64 The timestamp of when the request was made ResolveTime int64 The timestamp of when the last validator submitted the report and the request is resolved ResolveStatus int32 The resolve status of the request. Seehere for the full list of possible values Result []byte The aggregated value of the results returned by the validators

Requesting Data Through IBC

To make a request to BandChain's oracle using IBC, the module on another IBC-compatible blockchain must first initialize a communication tunnel with the oracle module on BandChain. Once the connection has been established, a pair of channel identifiers is generated -- one for the counterparty chain and one for BandChain.

The channel identifier is an important piece for the counterparty IBC module to route outgoing oracle request packets to the targeted oracle module on BandChain.

Similarly, BandChain's oracle module uses the channel identifier when sending back the oracle response. This means these channel identifiers have to be unique within each module, and the right channel identifier needs to be specified when making an oracle request from the counterparty chain.

Once a relayer has been set up, the module on another IBC-compatible blockchain looking to make the request must generate anOracleRequestPacketData data packet to be relayed. Using their chain's IBC module, they must then relay the message through to BandChain's own IBC module, which will proceed to further send it to the chain'soracle module. Once the request packet is successfully received, the subsequent flow is the almost the same as how BandChain handles a nativeMsgRequestData message type with a few additional steps. To summarize, the data request flow consists of the following steps:

- First, requesters create a request from their chain which are then relayed to BandChain.
- Once the request is submitted to BandChain, the oracle module fetches the corresponding oracle script and starts the oracle script's preparation phase returning information of all related data sources.
- Then BandChain performs various checks such as preparation phase smaller than the provided prepare gas (This is actually done in the oracle's script preparation phase) and the total fee for the request does not exceed the provided fee limit (More details can be found in the nextsection
-).

- (IBC Process Only) Then an acknowledgement is sent back to the requester's chain which either contains the error from the checks or the request identifier created by BandChain.
- If there is no error, the request is then broadcasted. Each validator selected for the particular request will then proceeed to retrieve data from each of the data source
- If a validator's retrieval is successful, they will submit back a report to BandChain containing the result they received from each of the data source.
- If the number of validators that managed to successfully submit the report exceeds theminCount
- specified in theOracleRequestPacketData
- , BandChain then computes and stores an aggregate final value.
- (IBC Process Only) The final result is also directly relayed back to the requesting chain and module in the form of aOracleResponsePacketData
- data packet.

As a slight aside, a data request to BandChain generally takes roughly 5 seconds from submitting the initial request until the requester received back the requested result. This is because BandChain's blocktime is set at approximately 3 seconds

Previous Protobuf Documentation Next Decentralized Validator Sampling