

Introduction

The [Cosmos SDK](#) is presently the most widely utilized framework for developing blockchain applications. A crucial necessity for such apps is an oracle. As decentralized applications (dApps) rely heavily on real-world data, ensuring the accuracy of this data holds significant importance.

It would be advantageous for your Cosmos SDK app to possess a built-in Oracle that obtains data from BandChain via IBC. By importing the pricefeed module implemented by Band Protocol.

Proposal

The initial step for the pricefeed module is to obtain information about the symbols that require price data from BandChain on everyn block. This is accomplished by submitting the `UpdateSymbolRequest` Proposal.

The Proposal submitted to update tasks for the pricefeed module consists of three components - the name of the symbol, the Oracle script ID required to obtain the price, and the block interval for requesting the data everyn blocks.

Upon the proposal's approval, the pricefeed module will request price data from BandChain based on the `SymbolRequest` that was updated through the proposal.

Workflow

Request

At a high level, the workflow will be as follows. First, the pricefeed module creates an IBC packet to request data from BandChain. Then, relayers will pick up the IBC packet and relay it on BandChain.

After BandChain processes the request, it will send an acknowledgement message along with `request_id` back. And, when the result of the request is finalized, BandChain will send a new IBC packet that contains the final data back. Relayers will listen and pick up those packets and relay them to your Cosmos SDK app.

After this stage, the Cosmos SDK app can safely use the data obtained from BandChain in its application at everyn blocks interval. If the Cosmos SDK app requires additional data, it can submit an update symbols request proposal at any time.

pricefeed

The pricefeed module obtains price data from BandChain through IBC and stores the most recent prices on your Cosmos SDK applications.

An example of the usage of this module is provided on the [Oracle Consumer Chain](#).

Params

The pricefeed module stores its params in the state, it can be updated with governance. The information contained in these parameters is utilized to request data from BandChain.

```
proto/consumer/pricefeed/params.proto message Params { option (gogoproto.goproto_stringer) = false;
```

```
uint64 ask_count = 1; uint64 min_count = 2; uint64 min_ds_count = 3; uint64 prepare_gas_base = 4; uint64  
prepare_gas_each = 5; uint64 execute_gas_base = 6; uint64 execute_gas_each = 7; string source_channel = 8; repeated  
cosmos.base.v1beta1.Coin fee_limit = 9 [ (gogoproto.nullable) = false, (gogoproto.castrepeated) =  
"github.com/cosmos/Cosmos SDK/types.Coins" ]; }
```

Proposal

The pricefeed module includes the `UpdateSymbolRequestProposal` for updating symbols to request price data on BandChain based on `block_interval` configuration by submitting the proposal on your Cosmos SDK application.

```
proto/consumer/pricefeed/pricefeed.proto message UpdateSymbolRequestProposal { option (gogoproto.goproto_getters) =  
false;
```

```
string title = 1; string description = 2;
```

```
repeated SymbolRequest symbol_requests = 3 [ (gogoproto.nullable) = false ]; }
```

message `SymbolRequest` { string symbol = 1; uint64 oracle_script_id = 2; uint64 block_interval = 3; } The example of submitting and voting on the proposal is demonstrated in the CLI section.

CLI

A user can query and interact with the pricefeed module using the CLI.

Note: This example uses `oracle-consumerd` as a command-line interface (CLI) from [oracle consumer chain](#). Please replace it with your own cosmos app.

Query

The query commands allow users to query pricefeed state.

```
oracle-consumerd query pricefeed --help
```

Symbol Requests

The `symbol-requests` command enables users to retrieve information about all symbol requests that are saved in this Cosmos SDK application.

```
oracle-consumerd query pricefeed symbol-requests
```

Price

The `price` command allows users to query price data by the symbol.

```
oracle-consumerd query pricefeed price [symbol] Example:
```

```
oracle-consumerd query pricefeed price BTC Example Output:
```

```
price: price: "22702955000000" resolve_time: "1675935544" symbol: BTC
```

Proposal

The `tx gov submit-legacy-proposal` commands allow users to submit a proposal on your Cosmos SDK app.

```
oracle-consumerd tx gov submit-legacy-proposal -h
```

Source Channel param change proposal

To acquire BandChain data through the IBC, it is imperative to update the `source-channel` parameter by submitting a proposal for the change that reflects your source channel.

```
oracle-consumerd tx gov submit-legacy-proposal param-change [proposal-file] Example:
```

1. `createparam_change.json`
2. file
3. {
4. "title"
5. :
6. "Param change for SourceChannel"
7. ,
8. "description"
9. :
10. "Proposal for change SourceChannel param in pricefeed module"
11. ,
12. "changes"
13. :
14. [
15. {
16. "subspace"
17. :
18. "pricefeed"
19. ,
20. "key"
21. :
22. "SourceChannel"
23. ,
24. "value"
25. :

```

26. "channel-0"
27. }
28. ]
29. ,
30. "deposit"
31. :
32. "10000000stake"
33. }
34. submit the proposal
35. oracle-consumerd tx gov submit-legacy-proposal param-change param_change.json --from alice

```

Update symbol request proposal

The `update-symbol-request` command allows users to update symbol requests to specify which symbols they desire to obtain price data from BandChain.

`oracle-consumerd tx gov submit-legacy-proposal update-symbol-request [proposal-file]` Example:

```

1. create update_symbol_requests.json
2. file
3. {
4. "title"
5. :
6. "Update Symbol requests"
7. ,
8. "description"
9. :
10. "Update symbol that request price from BandChain"
11. ,
12. "symbol_requests"
13. :
14. [
15. {
16. "symbol"
17. :
18. "BTC"
19. ,
20. "oracle_script_id"
21. :
22. "396"
23. ,
24. "block_interval"
25. :
26. "40"
27. }
28. ,
29. {
30. "symbol"
31. :
32. "ETH"
33. ,
34. "oracle_script_id"
35. :
36. "396"
37. ,
38. "block_interval"
39. :
40. "40"
41. }
42. ]
43. ,
44. "deposit"
45. :
46. "10000000stake"
47. }
48. Note: You can also delete symbol request by set "block_interval": "0"
49. on this proposal.
50. submit the proposal
51. oracle-consumerd tx gov submit-legacy-proposal update-symbol-request update_symbol_requests.json --from alice

```

Another way to initiate source channel and symbol requests

To utilize the Ignite feature to replace the genesis state, insert the code shown below into the `config.yml` file. and restart the chain by using `ignite chain serve -r -v` command.

```
genesis : app_state : pricefeed : params : source_channel :
```

```
'channel-0' symbol_requests :
```

```
[ {
```

```
'symbol' :
```

```
'BAND' ,
```

```
'oracle_script_id' :
```

```
396 ,
```

```
'block_interval' :
```

```
40
```

```
] ]
```

Learn more

- Cosmos SDK* [building-modules](#)
- - [ibc-app-packets](#)
- BandChain* [oracle-script](#) [Previous VRF Supported Blockchains](#) [Next Getting started](#)