

# Messages

## Instantiate

```
{ // A unique ID for the oracle request "client_id" :  
  
"cw-band-price-feed" , // The oracle script ID to query "oracle_script_id" :  
  
3 , // The number of validators that are requested to respond "ask_count" :  
  
1 , // The minimum number of validators that need to respond "min_count" :  
  
1 , // The maximum amount of band in uband to be paid to the data source providers // e.g. vec![Coin::new(100, "uband")]  
"fee_limit" :  
  
[ { "amount" : "100000" , "denom" : "uband" } ] , // Amount of gas to pay to prepare raw requests "prepare_gas" :  
  
1000 , // Amount of gas reserved for execution "execute_gas" :  
  
5000 , // Minimum number of sources required to return a successful response "multiplier" :  
  
1000000 , // The list of symbols to query "symbols" :  
  
[ "ATOM" ,  
  
"USDC" ] , // The owner of the contract "owner" :  
  
"neutron..." , // The maximum time interval between updates in seconds "max_update_interval" :  
  
"100" }
```

## Execute

### request

```
{ "request" :  
  
{ } } Instantiates a request for a new price data from Band Protocol.
```

### update\_config

```
{ "update_config" :  
  
{ "new_config" :  
  
{ // A unique ID for the oracle request "client_id" :  
  
"cw-band-price-feed" , // The oracle script ID to query "oracle_script_id" :  
  
3 , // The number of validators that are requested to respond "ask_count" :  
  
1 , // The minimum number of validators that need to respond "min_count" :  
  
1 , // The maximum amount of band in uband to be paid to the data source providers // e.g. vec![Coin::new(100, "uband")]  
"fee_limit" :  
  
[ { "amount" : "100000" , "denom" : "uband" } ] , // Amount of gas to pay to prepare raw requests "prepare_gas" :  
  
1000 , // Amount of gas reserved for execution "execute_gas" :  
  
5000 , // Minimum number of sources required to return a successful response "multiplier" :  
  
1000000 , // The list of symbols to query "symbols" :  
  
[ "ATOM" ,
```

```
"USDC" ] , // The maximum time interval between updates in seconds "max_update_interval" :  
"100" } } } Owner only method. Updates contract's configuration.
```

## **update\_owner**

```
{ "update_owner" :
```

```
{ "new_owner" :
```

```
"neutron..." } } Owner only method. Updates contract's owner.Previous Overview Next Queries
```