

Capital Call

Asset Type

This is basically an enum which is used to perform different operations that can be either of the following

1. Token
2. Stable Coin
3. Fiat

When **asset_type = Token**, then token related operations will be performed

When **asset_type = Stable Coin**, then stable coins related operations will be performed

When **asset_type = Fiat**, then Fiat related operations will be performed

Coin Type

This is basically an enum which is used to switch between different types of stable coins. Currently for testing we are use following coins but later on these coins will be replaced by Provenance chain based stable coins like omni, axlUSDC, etc

1. USDC-Test
2. USDT-Test
3. DAI-Test

When **coin_type = USDC-Test**, transactions will be taken place on USDC-Test coins and user must have these coins for the transaction to happen

When **coin_type = USDT-Test**, transactions will be taken place on USDT-Test coins and user must have these coins for the transaction to happen

When **coin_type = DAI-Test**, transactions will be taken place on DAI-Test coins and user must have these coins for the transaction to happen

Notes: For Gas Fees, nHash will be required, only for transfer of stable coins above respective coin-type will be used

Update Type

This is basically an enum which is used to manage admins, agents and user management fees. There are three kinds of the same:-

1. Add
2. Remove
3. Update

When **update_type = Add**, use to perform add operations

When **update_type = Remove**, use to perform remove operations

When **update_type = Update**, use to perform update operations

This update_type is of generic type, so multiple kinds of data types are handled in the smart contract such as address, arrays, structure, etc,

Create

This function is used to create new funds for a particular token. Anyone can call this function and the caller will be assigned as the agent. Here the contract address will be assigned as the agent of the token contract to perform mint, burn, transfer and force transfer operations.

Parameters

```
{
  "create": {
    "params": {
      "denom": "string",
      "fund_name": "string",
      "asset_type": "string",
      "issuer_name": "string",
      "target_aum": "Uint128",
      "nav_launch_price": "Uint128",
      "ccy": "string"
    }
  }
}
```

Waterfall distribution (Share dividend)

This function can be called only by the agent. This function supports batch stable coin transfers and batch minting. The stable coins will be selected from the coin_type while operations will be selected based on asset_type.

When **asset_type = Token**, then tokens will be minted to the recipient wallet.

When **asset_type = Stable Coin**, then stable coins will be transferred from the agent's wallet to the recipient's wallet.

Parameters

```
{
  "share_dividend": {
    "denom": "string",
    "coin_type": "enum",
```

```

    "shared_dividends": [
      {
        "to": "address",
        "dividend": "Uint128",
        "asset_type": "enum"
      },
      {
        "to": "address",
        "dividend": "Uint128",
        "asset_type": "enum"
      }
    ]
  }
}

```

Distribute and Burn

This function can be called only by the agent. This function supports batch stable coin transfers and batch burning. The stable coins will be selected from the coin_type. When this function is called, the amount of tokens will be transferred from the investor's account and different amounts of stable coins will be transferred from the agent's account to the investor's account.

Parameters

```

{
  "distribute_and_burn": {
    "denom": "string",
    "coin_type": "enum",
    "distributions": [
      {
        "investor": "address",
        "amount": "Uint128",
        "token": "Uint128"
      },
      {
        "investor": "address",
        "amount": "Uint128",
        "token": "Uint128"
      }
    ]
  }
}

```

Send Stable Coins

With the help of this function the amount of stable coins will be transferred from the sender's wallet to the agent's wallet which is associated by a unique token. This function can be called by anyone.

Parameters

```
{
  "send_stable_coins": {
    "denom": "string"
  }
}
```

Rescue Tokens

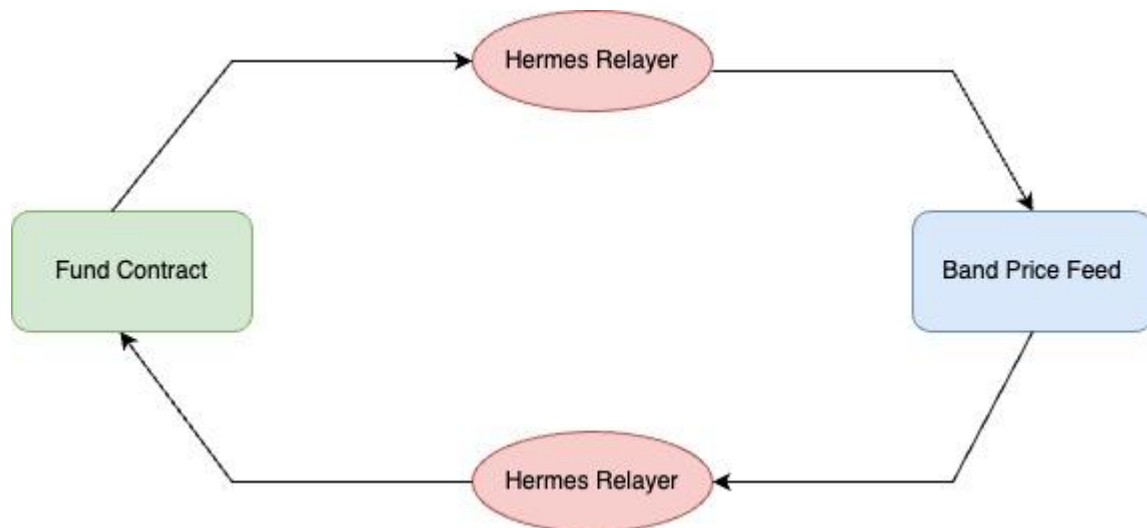
With the help of this function, any extra tokens will be transferred back to the recipient's wallet. This function can be called only by the agent.

Parameters

```
{
  "send_stable_coins": {
    "denom": "string"
  }
}
```

Fetch Price

This function is used to fetch the price of the given currency from the on-chain data. This function can be called only by the admin. The relayer is running in the background that will collect the latest price from the on-chain. Here band protocol is used to collect the price and hermes is used as relayer.



Parameters

```
{
  "fetch_price": {
    "denom": "string"
  }
}
```

Manage Admins

This function is used to add or remove addresses as admin. This function can be called only by the admin.

Parameters

```
{
  "manage_admins": {
    "update_type": {
      "<add or remove>": [
        "address"
      ]
    }
  }
}
```

Manage Agents

This function is used to add or remove addresses as agents. This function can be called only by the admin.

Parameters

```
{
  "manage_agent": {
    "denom": "string",
    "update_type": {
      "<add, remove or update>": "address"
    }
  }
}
```

Management Fees

This function is used to add, remove or update users' management fees. This function can be called only by the agent.

Parameters

```
{
  "management_fees": {
    "denom": "string",
    "managed_users": [
      {
        "<add or remove>": {
          "user": "address",
          "fees": "Uint128"
        }
      }
    ]
  }
}
```

```
    }  
  }  
]  
}  
}
```

Update Currency

This function is used to update the currency which is being used by the relayer to fetch price from on-chain data. This function can be called only by the admin.

Parameters

```
{  
  "update_currency": {  
    "denom": "string",  
    "ccy": "string"  
  }  
}
```