hack-sdk-go

Golang version SDK for hackathon

Requirements

- Go 1.12 or higher.

Installation

Auto installed by go module (recommended):
Go module will install it if you import the package in your code.

Manually install by go get:

```
go get "github.com/annchain/hack-sdk-go"
```

Usage

To get started with the SDK, import the package and create an OgSolver object:

```
import (
    hackSDK "github.com/annchain/hack-sdk-go"
)

func main() {
     url := "http://localhost:8000"
     kafkaUrl := "localhost:9092"
     // the private key of the account you want to send the tx.
     priv :=
"0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef"
     token := "we will offer token for each team"

     ogSolver, err := hackSDK.NewOgSolver(url, kafkaUrl, priv, token)
     if err != nil {
      // handle the error
     }
}
```

Query the balance, use the OgSolver object created above:

```
// query balance
addressToQuery := "0x0123456789abcdef0123456789abcdef01234567"
balance, err := ogSolver.QueryBalance(addressToQuery)
if err != nil {
    fmt.Println("query balance error: ", err)
    return
}
fmt.Println("balance is: ", balance)
```

Send a transaction:

```
import (
    "math/big"
)

// query nonce
nonce, err := ogSolver.QueryNonce(ogSolver.Address())
if err != nil {
    fmt.Println("query nonce error: ", err)
    return
}

// find parents, you should find proper parents by yourself
parents := []string{ "0xf9733413...", "0xa857bb7f..." }

// create transaction
tx := hackSDK.Transaction{}
tx.Parents = parents
tx.From = ogSolver.Address()
tx.Nonce = nonce + 1
tx.Guarantee = big.NewInt(100)  // the guarantee you want to bet

// send tx
hash, err := ogSolver.SendTx(tx)
if err != nil {
    fmt.Println("send tx error: ", err)
    return
}
fmt.Println("tx hash is: ", hash)
```

Constantly get new transactions:

```
receiver := ogSolver.ReceiveNewestTx()

for {
    fmt.Println("start consuming one data")
    select {
    case txiResp := <-receiver:
```

```
        if txiResp.Type == hackSDK.TxTypeNormal {
            txResp := txiResp.Data.(hackSDK.TransactionResp)
            // do something if you get a new transaction.
        }
        if txiResp.Type == hackSDK.TxTypeSequencer {
            seqResp := txiResp.Data.(hackSDK.SequencerResp)
            // do something if you get a new sequencer.
        }

    }
}
```

Query the infomation of next sequencer:

```
data, err := og.QueryNextSequencerInfo()
if err != nil {
    // handle the error
    return
}

fmt.Println(data)
```