

Records

[Suggest Edits](#)

The Records Module

The records module handles record keeping and accounting for the Stride blockchain.

It is [IBC middleware](#) . IBC middleware wraps core IBC modules and other middlewares. Specifically, the records module adds a middleware stack to app.go with the following structure: records -> transfer . All ibc packets routed to the transfer module will first pass through records , where we can apply custom logic (record keeping) before passing messages to the underlying transfer module.

Note:

- The middleware stack is added in
- app.go
- The custom handler logic is added in
- ibc_module.go
- by implementing the IBCModule interface

Keeper functions

Deposit Records

- GetDepositRecordCount()
- SetDepositRecordCount()
- AppendDepositRecord()
- SetDepositRecord()
- GetDepositRecord()
- RemoveDepositRecord()
- GetAllDepositRecord()
- GetTransferDepositRecordByEpochAndChain()

Epoch Unbonding Records

- SetEpochUnbondingRecord()
- GetEpochUnbondingRecord()
- RemoveEpochUnbondingRecord()
- GetAllEpochUnbondingRecord()
- GetAllPreviousEpochUnbondingRecords()
- GetHostZoneUnbondingByChainId()
- AddHostZoneToEpochUnbondingRecord()
- SetHostZoneUnbondingStatus()

User Redemption Records

- SetUserRedemptionRecord()
- GetUserRedemptionRecord()
- RemoveUserRedemptionRecord()
- GetAllUserRedemptionRecord()
- IterateUserRedemptionRecords()

State

Callbacks

- TransferCallback

Genesis

- UserRedemptionRecord
- Params
- RecordsPacketData
- NoData
- DepositRecord
- HostZoneUnbonding

- EpochUnbondingRecord
- GenesisState

Queries

- Params
- GetDepositRecord
- AllDepositRecord
- GetUserRedemptionRecord
- AllUserRedemptionRecord
- AllUserRedemptionRecordForUser
- GetEpochUnbondingRecord
- AllEpochUnbondingRecord

Events

The records module emits does not currently emit any events. Updated 2 months ago

[StakeIBC Claim](#) Did this page help you? Yes No * [Table of Contents](#) * * [The Records Module](#) * * * [Keeper functions](#) * * * [State](#) * * * [Queries](#) * * * [Events](#)