

# Accounts and Wallets

Recommended Readings \* [Cosmos SDK Accounts](#)

## Band Accounts

BandChain is a Cosmos SDK-based blockchain, which means it is compatible with the Cosmos SDK account system. However, BandChain has its own customAccount type to implement in Hierarchical Deterministic (HD) wallets.

The BandChain coin type HD path is m/44'/494'/0'/0/0. This path is used to derive BandChain addresses from [BIP44 standard](#) HD wallet.

The first part of the path, m/44', specifies the derivation path is based on BIP44 standard. The second part of the path, 494', indicates the BandChain is the coin type for this address. The third part of the path, 0', specifies that this is a hardened derivation path, which means that the child keys can only be derived by someone who has the parent key. The fourth part of the path, 0, specifies the account index which is used to generate multiple accounts from the same master key. The last path, 0, specifies the address index, which is used to generate multiple addresses for a single account.

## Addresses and Public Keys

BandChain supports [Bech32](#) (band...). The Bech32 format is the default format for Cosmos-SDK queries and transactions through CLI and REST clients.

For example, a BandChain address might look like this: band18p27yl962l8283ct7srr5l3g7ydazj07dqrwph. This address can be used to receive and send BandChain tokens and interact with smart contracts and dApps on the BandChain network.

Addresses and PubKeys are both public information that identifies actors in the application. Each account is identified using Address which is a sequence of bytes derived from a public key.

In the BandChain, we define 3 types of addresses that specify a context where an account is used:

Types Address bech32 Prefix Pubkey bech32 Prefix Address byte length Pubkey byte length Accounts band bandpub 20 33 Validators bandvaloper bandvaloperpub 20 33 Consensus Nodes bandvalcons bandvalconspub 20 33

## Mnemonic and Address Generation

To create an account you can either create a private key, a keystore file (a file that contains your encrypted private key), or a mnemonic phrase (a string of words that can access multiple private keys).

Mnemonic phrases also known as [hierarchical deterministic key generation](#) (HD keys). This allows the user to create accounts on multiple blockchains without having to manage multiple secrets. We can think of the derived accounts as child accounts of the root account created using the original mnemonic seed phrase.

A mnemonic phrase typically consists of 12, 24, or sometimes 16 words that are randomly generated from a pre-determined word list. Each word in the list corresponds to a unique sequence of numbers, which are used to generate the private key for your wallet. By writing down and securely storing your mnemonic phrase, you can always recover your BandChain account in case of loss or damage to your device.

## Example of mnemonic phrases used in BandChain

hedgehog pact leave raccoon empty various item profit uncover dune someone ball chat repair acquire middle error rally isolate group hair replace buzz survey Protect your seed phrase If the mnemonic phrase leaks, accounts cannot be derived without the initial password. If someone knows the mnemonic phrase and the derivation path, they will have access to your account. To protect your seed phrase, consider the following tips.

- Never share your mnemonic phrase with anyone.
- Always write down your seed phrase with a pen and paper.
- Store the paper with your seed phrase on it somewhere safe.
- Never give your seed phrase to anyone, not even support staff. At BandChain, we offer a simple method for generating a mnemonic phrase to create a BandChain account using either [PyBand](#) or [BandChain.js](#).

```
import
```

```
{
```

```
Wallet
```

```

}

from

"@bandprotocol/bandchain.js" ; const

{

PrivateKey

}

=

Wallet

const

[ mnemonic , privateKey ]

=

PrivateKey . generate ( "m/44'/494'/0'/0/0" ) const address =

PrivateKey . fromMnemonic ( mnemonic ) . toPubkey ( ) . toAccBech32 ( )

// band1ycw2277nurr5zymw7exqf8za2t73y3ys5zwf7z

```

## Querying an Account

You can query an account address using the Band CLI or REST clients:

## Query account address using bandd

bandd keys show < key-name

-a Query account address using REST client

## GET /cosmos/auth/v1beta1/accounts/{address}

```

curl -X GET
"https://laozi1.bandchain.org/api/cosmos/auth/v1beta1/accounts/band1ycs4g7xu8wmf7n4vwwtfsvhtfm7tekw683qrf" -H
"accept: application/json"

```

## Wallets

As mentioned earlier, building on the CosmosSDK allows us to enable our users to interact with BandChain using Cosmos native wallets. The most popular Cosmos and IBC-enabled wallets are supported on BandChain. These include:

1. [Cosmostation](#)
2. [Ledger Previous Gas and Fees Next Protocol Messages](#)