Wallets and private keys

Introduction

Wallets allow individuals to manage, store and transact across blockchain networks. Wallets are composed of a public and private key pair. The private key is a unique and secret cryptographic code that provides ownership and control over the wallet and allows individuals to authorize transactions signing. The public address is a cryptographic identifier that allows individuals to receive and view transactions.

Changes to the state of a network are achieved by broadcasting transactions. To accomplish this, we can use CosmPy in a straightforward way to create and manage private keys and addresses. In this guide, you will learn how to both generate anew private key as well asrecover a previously generated one .

The examples below outlines how to achieve both of these operations.

Generate a new private key

First of all, import the needed modules and then use thePrivateKey claass to create yourprivate_key :

Import necessary classes

```
from cosmpy . aerial . wallet import LocalWallet from cosmpy . crypto . keypairs import PrivateKey
```

Creating a random private key

private_key

```
PrivateKey ()
```

Recover an existing private key

Let's start by extracting the private key and convert it into a base64 encoded string. You can do this on macOS or Linux for the Fetch.ai network using the FetchD CL.

An example is provided in the code snippet below:

Extract the private key and convert it into a base64 encoded string

fetchd keys export mykeyname --unsafe --unarmored

```
hex
```

| xxd - r - p | base64 Once you have extracted and converted the key into a base64 encoded string you can recover to an existing private key by following along the provided example below:

Import necessary classes

```
from cosmpy . aerial . wallet import LocalWallet from cosmpy . crypto . keypairs import PrivateKey
```

Recovering an existing private key

private_key

```
PrivateKey ( " )
```

Here is where you provide the base64 encoded private key string

ThePrivateKey object is one of CosmPy's low level primitives. This is why it is generally paired with aWallet object. You can now create a local wallet using theLocalWallet class from the CosmPy library and initialize it with theprivate_key you generated or recovered earlier. TheLocalWallet class is used to manage a local wallet associated with a private key for interacting with the Cosmos blockchain.

With this wallet object, you can perform various operations, such as generating the corresponding public address associated with the private key and signing transactions. You can also query useful information such as the address from the wallet directly. The example below showcases how to achieve both of these outlined operations:

wallet

LocalWallet (private_key) print (wallet. address ())

will print the address for the wallet

From mnemonic

You can also use an account's mnemonic phrase to get the associated private key. The example provided below showcases how to achieve that:

from cosmpy . aerial . wallet import LocalWallet

mnemonic

"person knife december tail tortoise jewel warm when worry limit reward memory piece cool sphere kitchen knee embody soft own victory sauce silly page"

wallet

LocalWallet . from_mnemonic (mnemonic)

Was this page helpful?

[Querying balances](#)   [Sending funds](#)