
py-stellar-base Documentation

Release 7.0.0

Stellar Community

Jan 12, 2022

CONTENTS

1 Quickstart	3
1.1 Installation	3
1.2 Generate Keypair	3
1.3 Create Account	4
1.4 Querying Horizon	6
1.5 Assets	7
1.6 Building Transactions	8
1.7 Creating a payment transaction	10
1.8 Asynchronous	13
1.9 Multi-signature account	16
1.10 XDR	18
2 API Documentation	21
2.1 API Documentation	21
3 stellar-model	257
4 Links	259
5 Thanks	261
6 genindex	263
Python Module Index	265
Index	267

py-stellar-base is a Python library for communicating with a [Stellar Horizon server](#). It is used for building Stellar apps on Python. It supports **Python 3.6+** as well as PyPy 3.6+.

It provides:

- a networking layer API for Horizon endpoints.
- facilities for building and signing transactions, for communicating with a Stellar Horizon instance, and for submitting transactions or querying network history.

QUICKSTART

At the absolute basics, you'll want to read up on [Stellar's Documentation Guides](#), as it contains a lot of information on the concepts used below (Transactions, Payments, Operations, KeyPairs, etc.).

1.1 Installation

1.1.1 Via pip

Use pip to install and update py-stellar-base:

```
pip install -U stellar-sdk
```

The py-stellar-base release follows [Semantic Versioning 2.0.0](#), and I strongly recommend that you specify its major version number in the dependency file to avoid the unknown effects of a corrupt update. More on installing Python and dependencies can be found over in the [Hitchhiker's Guide to Python](#).

1.1.2 Via Source Code

Please use the code on pypi whenever possible. The latest code may be unstable.

You can clone [the repository](#) directly, and install it locally:

```
git clone https://github.com/StellarCN/py-stellar-base.git
cd py-stellar-base
pip install .
```

1.2 Generate Keypair

The *Keypair* object represents a key pair used to sign transactions in a Stellar network. The *Keypair* object can contain both a public and a private key, or only a public key.

If a *Keypair* object does not contain a private key it can't be used to sign transactions. The most convenient method of creating a new keypair is by passing the account's secret seed:

```
1 from stellar_sdk import Keypair
2
3 secret = "SBK2VIYYSVG76E7VC3QHYARNFLY2EAQXDHRC7BMXBBGIFG74ARPRMNQM"
4 keypair = Keypair.from_secret(secret)
```

(continues on next page)

(continued from previous page)

```

5
6 # GDHmw6QzOL73SHKG2JA3YHXFDHM46SS5ZRWEYF5BCYHX2C5TV06KZBYL
7 public_key = keypair.public_key
8
9 can_sign = keypair.can_sign() # True

```

You can create a keypair from public key, but its function is limited:

```

1 from stellar_sdk import Keypair
2
3 public_key = "GDHmw6QzOL73SHKG2JA3YHXFDHM46SS5ZRWEYF5BCYHX2C5TV06KZBYL"
4 keypair = Keypair.from_public_key(public_key)
5 can_sign = keypair.can_sign() # False

```

You can also create a randomly generated keypair:

```

1 from stellar_sdk import Keypair
2
3 keypair = Keypair.random()
4 print("Public Key: " + keypair.public_key)
5 print("Secret Seed: " + keypair.secret)

```

1.3 Create Account

Now, in order to create an account, you need to run a `CreateAccount` operation with your new account ID. Due to Stellar's `minimum account balance`, you'll need to transfer the minimum account balance from another account with the create account operation. As of this writing, minimum balance is **1 XLM (2 x 0.5 Base Reserve)**, and is subject to change.

1.3.1 Using The SDF Testnet

If you want to play in the Stellar test network, you can ask our `Friendbot` to create an account for you as shown below:

```

1 """
2 This example shows how to activate an account via friendbot in a test network.
3
4 This feature is only available for test networks.
5
6 See: https://developers.stellar.org/docs/tutorials/create-account/#create-account
7 """
8 import requests
9
10 from stellar_sdk import Keypair
11
12 keypair = Keypair.random()
13
14 print("Public Key: " + keypair.public_key)
15 print("Secret Seed: " + keypair.secret)
16

```

(continues on next page)

(continued from previous page)

```

17 url = "https://friendbot.stellar.org"
18 response = requests.get(url, params={"addr": keypair.public_key})
19 print(response)

```

1.3.2 Using The Stellar Live Network

On the other hand, if you would like to create an account on the live network, you should buy some Stellar Lumens from an exchange. When you withdraw the Lumens into your new account, the exchange will automatically create the account for you. However, if you want to create an account from another account of your own, here's an example of how to do so:

```

1 """
2 This example shows how to create and fund a new account with the specified starting_
3 ↪balance.
4
5 See: https://developers.stellar.org/docs/tutorials/create-account/#create-account
6 See: https://developers.stellar.org/docs/start/list-of-operations/#create-account
7 """
8
9 from stellar_sdk import Keypair, Network, Server, TransactionBuilder
10
11 server = Server(horizon_url="https://horizon-testnet.stellar.org")
12 source = Keypair.from_secret("SBFZCHU5645DOKRWYBXVOXY2ELGJKFRX6VGGPRYUWHQ7PMXXJNDZFMKD")
13 destination = Keypair.random()
14
15 source_account = server.load_account(account_id=source.public_key)
16 transaction = (
17     TransactionBuilder(
18         source_account=source_account,
19         network_passphrase=Network.TESTNET_NETWORK_PASSPHRASE,
20         base_fee=100,
21     )
22     .append_create_account_op(
23         destination=destination.public_key, starting_balance="12.25"
24     )
25     .set_timeout(30)
26     .build()
27 )
28 transaction.sign(source)
29 response = server.submit_transaction(transaction)
30 print(f"Transaction hash: {response['hash']}")
31 print(
32     f"New Keypair: \n\taccount id: {destination.public_key}\n\tsecret seed: {destination.
33     ↪secret}"
34 )

```

Note: To avoid risks, TESTNET is used in the example above. In order to use the Stellar Live Network you will have to change the network passphrase to `Network.PUBLIC_NETWORK_PASSPHRASE` and the server URL to point to `https://horizon.stellar.org` too.

1.4 Querying Horizon

py-stellar-base gives you access to all the endpoints exposed by Horizon.

1.4.1 Building requests

py-stellar-base uses the [Builder pattern](#) to create the requests to send to Horizon. Starting with a [Server](#) object, you can chain methods together to generate a query. (See the [Horizon reference](#) documentation for what methods are possible.)

```
1 """
2 See: https://stellar-sdk.readthedocs.io/en/latest/querying\_horizon.html#building-requests
3 """
4 from stellar_sdk import Server
5
6 server = Server(horizon_url="https://horizon.stellar.org")
7 account = "GB6NVEN5HSUBKMYCE5ZOWSK5K23TBWRUQLZY3KNMXUZ3AQ2ESC4MY4AQ"
8
9 # get a list of transactions that occurred in ledger 1400
10 transactions = server.transactions().for_ledger(1400).call()
11 print(transactions)
12
13 # get a list of transactions submitted by a particular account
14 transactions = server.transactions().for_account(account_id=account).call()
15 print(transactions)
16
17 # The following example will show you how to handle paging
18 print(f"Gets all payment operations associated with {account}.")
19 payments_records = []
20 payments_call_builder = (
21     server.payments().for_account(account).order(desc=False).limit(10)
22 ) # limit can be set to a maximum of 200
23 payments_records += payments_call_builder.call()[ "_embedded" ][ "records" ]
24 page_count = 0
25 while page_records := payments_call_builder.next()[ "_embedded" ][ "records" ]:
26     payments_records += page_records
27     print(f"Page {page_count} fetched")
28     print(f"data: {page_records}")
29     page_count += 1
30 print(f"Payments count: {len(payments_records)}")
```

Once the request is built, it can be invoked with `call()` or with `stream()`. `call()` will return the response given by Horizon.

1.4.2 Streaming requests

Many requests can be invoked with `stream()`. Instead of returning a result like `call()` does, `stream()` will return an `EventSource`. Horizon will start sending responses from either the beginning of time or from the point specified with `cursor()`. (See the [Horizon reference](#) documentation to learn which endpoints support streaming.)

For example, to log instances of transactions from a particular account:

```

1  """
2  See: https://stellar-sdk.readthedocs.io/en/latest/querying\_horizon.html#streaming-
↳requests
3  """
4  from stellar_sdk import Server
5
6  server = Server(horizon_url="https://horizon-testnet.stellar.org")
7  account_id = "GASOCNHNLYFNMDJYQ3XFMI7BYHIOCFW3GJEOWRPEGK2TDPGTG2E5EDW"
8  last_cursor = "now" # or load where you left off
9
10
11 def tx_handler(tx_response):
12     print(tx_response)
13
14
15 for tx in server.transactions().for_account(account_id).cursor(last_cursor).stream():
16     tx_handler(tx)

```

1.5 Assets

Object of the `Asset` class represents an asset in the Stellar network. Right now there are 3 possible types of assets in the Stellar network:

- native XLM asset (`ASSET_TYPE_NATIVE`),
- issued assets with asset code of maximum 4 characters (`ASSET_TYPE_CREDIT_ALPHANUM4`),
- issued assets with asset code of maximum 12 characters (`ASSET_TYPE_CREDIT_ALPHANUM12`).

To create a new native asset representation use static `native()` method:

```

1  from stellar_sdk import Asset
2  native = Asset.native()

```

To represent an issued asset you need to create a new object of type `Asset` with an asset code and issuer:

```

1  from stellar_sdk import Asset
2  # Creates TEST asset issued by GBBM6BKZPEHWYO3E3YKREDPQXMS4VK35YLNUN7NFBRI26RAN7GI5POFBB
3  test_asset = Asset("TEST", "GBBM6BKZPEHWYO3E3YKREDPQXMS4VK35YLNUN7NFBRI26RAN7GI5POFBB")
4  is_native = test_asset.is_native() # False
5  # Creates Google stock asset issued by
↳GBBM6BKZPEHWYO3E3YKREDPQXMS4VK35YLNUN7NFBRI26RAN7GI5POFBB
6  google_stock_asset = Asset('US38259P7069',
↳'GBBM6BKZPEHWYO3E3YKREDPQXMS4VK35YLNUN7NFBRI26RAN7GI5POFBB')
7  google_stock_asset_type = google_stock_asset.type # credit_alphanum12

```

1.6 Building Transactions

Transactions are the commands that modify the state of the ledger. They include sending payments, creating offers, making account configuration changes, etc.

Every transaction has a source **account**. This is the account that pays the **fee** and uses up a sequence number for the transaction.

Transactions are made up of one or more **operations**. Each operation also has a source account, which defaults to the transaction's source account.

1.6.1 TransactionBuilder

The *TransactionBuilder* class is used to construct new transactions. TransactionBuilder is given an account that is used as transaction's **source account**. The transaction will use the current sequence number of the given *Account* object as its sequence number and increments the given account's sequence number when *build()* is called on the TransactionBuilder.

Operations can be added to the transaction calling *append_operation* for each operation you wish to add to the transaction. See *Operation* for a list of possible operations you can add. *append_operation* returns the current *TransactionBuilder* object so you can chain multiple calls. But you don't need to call it directly, we have prepared a lot of easy-to-use functions for you, such as *append_payment_op* can add a payment operation to the *TransactionBuilder*.

After adding the desired operations, call the *build()* method on the *TransactionBuilder*. This will return a fully constructed *TransactionEnvelope*. The transaction object is wrapped in an object called a *TransactionEnvelope*, the returned transaction will contain the sequence number of the source account. This transaction is unsigned. You must sign it before it will be accepted by the Stellar network.

```

1  """
2  This example demonstrates how to use TransactionBuilder
3  to quickly build a transaction. For a beginner,
4  most of the work can be done with TransactionBuilder.
5
6  See: https://stellar-sdk.readthedocs.io/en/latest/building\_transactions.html#building-
8  ↪transactions
9  """
10 from stellar_sdk import Account, Asset, Keypair, Network, TransactionBuilder
11
12 root_keypair = Keypair.from_secret(
13     "SA6XHAH4GNLRWWWF6TEVEWNS44CBNFAJWHWOPZCVZOUXSQA7BOYN7XHC"
14 )
15 # Create an Account object from an address and sequence number.
16 root_account = Account(account=root_keypair.public_key, sequence=1)
17
18 transaction = (
19     TransactionBuilder(
20         source_account=root_account,
21         # If you want to submit to pubnet, you need to change `network_passphrase` to ↪
22         ↪`Network.PUBLIC_NETWORK_PASSPHRASE`
23         network_passphrase=Network.TESTNET_NETWORK_PASSPHRASE,
24         base_fee=100,
25     )
26     .append_payment_op( # add a payment operation to the transaction

```

(continues on next page)

(continued from previous page)

```

24     destination="GASOCNHNLYFNMDJYQ3XFMI7BYHIOCFW3GJEOWRPEGK2TDPGTG2E5EDW",
25     asset=Asset.native(),
26     amount="125.5",
27 )
28 .append_set_options_op( # add a set options operation to the transaction
29     home_domain="overcat.me"
30 )
31 .set_timeout(30)
32 .build()
33 ) # mark this transaction as valid only for the next 30 seconds

```

1.6.2 Sequence Numbers

The sequence number of a transaction has to match the sequence number stored by the source account or else the transaction is invalid. After the transaction is submitted and applied to the ledger, the source account's sequence number increases by 1.

There are two ways to ensure correct sequence numbers:

1. Read the source account's sequence number before submitting a transaction
2. Manage the sequence number locally

During periods of high transaction throughput, fetching a source account's sequence number from the network may not return the correct value. So, if you're submitting many transactions quickly, you will want to keep track of the sequence number locally.

1.6.3 Adding Memos

Transactions can contain a **memo** field you can use to attach additional information to the transaction. You can do this by passing a *Memo* object when you construct the *TransactionBuilder*.

There are 5 types of memos:

- *stellar_sdk.memo.NoneMemo* - empty memo,
- *stellar_sdk.memo.TextMemo* - 28-byte ascii encoded string memo,
- *stellar_sdk.memo.IdMemo* - 64-bit number memo,
- *stellar_sdk.memo.HashMemo* - 32-byte hash - ex. hash of an item in a content server,
- *stellar_sdk.memo.ReturnHashMemo* - 32-byte hash used for returning payments - contains hash of the transaction being rejected.

```

1  """
2  This example shows how to add memo to a transaction.
3
4  See: https://developers.stellar.org/docs/glossary/transactions/#memo
5  See: https://stellar-sdk.readthedocs.io/en/latest/building\_transactions.html#building-
6  ↪ transactions
7  """
8  from stellar_sdk import Account, Asset, Keypair, Network, TransactionBuilder
9  root_keypair = Keypair.from_secret(

```

(continues on next page)

(continued from previous page)

```

10     "SA6XHAH4GNLRWWWF6TEVEWNS44CBNFAJWHWOPZCVZOUXSQA7BOYN7XHC"
11 )
12 # Create an Account object from an address and sequence number.
13 root_account = Account(account=root_keypair.public_key, sequence=1)
14
15 transaction = (
16     TransactionBuilder(
17         source_account=root_account,
18         network_passphrase=Network.TESTNET_NETWORK_PASSPHRASE,
19         base_fee=100,
20     )
21     .add_text_memo("Happy birthday!")
22     .append_payment_op(
23         destination="GASOCNHNLYFNMDJYQ3XFMI7BYHIOCFW3GJEOWRPEGK2TDPGTG2E5EDW",
24         amount="2000",
25         asset=Asset.native(),
26     )
27     .set_timeout(30)
28     .build()
29 )

```

1.6.4 Transaction and TransactionEnvelope

These two concepts may make the novices unclear, but the official has given a good explanation.

Transactions are commands that modify the ledger state. Among other things, Transactions are used to send payments, enter orders into the distributed exchange, change settings on accounts, and authorize another account to hold your currency. If you think of the ledger as a database, then transactions are SQL commands.

Once a transaction is ready to be signed, the transaction object is wrapped in an object called a Transaction Envelope, which contains the transaction as well as a set of signatures. Most transaction envelopes only contain a single signature along with the transaction, but in multi-signature setups it can contain many signatures. Ultimately, transaction envelopes are passed around the network and are included in transaction sets, as opposed to raw Transaction objects.

1.7 Creating a payment transaction

1.7.1 Payment

In this example, the destination account must exist. We use synchronous methods to submit transactions here, if you want, you can also use asynchronous methods.

```

1  """
2  Create, sign, and submit a transaction using Python Stellar SDK.
3
4  Assumes that you have the following items:
5  1. Secret key of a funded account to be the source account
6  2. Public key of an existing account as a recipient
7     These two keys can be created and funded by the friendbot at
8     https://www.stellar.org/laboratory/ under the heading "Quick Start: Test Account"
9  3. Access to Python Stellar SDK (https://github.com/StellarCN/py-stellar-base) through
    ↪ Python shell.

```

(continues on next page)

(continued from previous page)

```

10
11 See: https://developers.stellar.org/docs/start/list-of-operations/#payment
12 """
13 from stellar_sdk import Asset, Keypair, Network, Server, TransactionBuilder
14
15
16 def create_account():
17     """To make this script work, create an account on the testnet."""
18     import requests
19
20     from stellar_sdk import Keypair
21
22     keypair = Keypair.random()
23     url = "https://friendbot.stellar.org"
24     _response = requests.get(url, params={"addr": keypair.public_key})
25     # Check _response.json() in case something goes wrong
26     return keypair
27
28
29 # The source account is the account we will be signing and sending from.
30 example_keypair = create_account()
31 source_secret_key = example_keypair.secret
32
33 # Derive Keypair object and public key (that starts with a G) from the secret
34 source_keypair = Keypair.from_secret(source_secret_key)
35 source_public_key = source_keypair.public_key
36
37 # We just send lumen to ourselves in this simple example
38 receiver_public_key = example_keypair.public_key
39
40 # Configure StellarSdk to talk to the horizon instance hosted by Stellar.org
41 # To use the live network, set the hostname to 'horizon.stellar.org'
42 server = Server(horizon_url="https://horizon-testnet.stellar.org")
43
44 # Transactions require a valid sequence number that is specific to this account.
45 # We can fetch the current sequence number for the source account from Horizon.
46 source_account = server.load_account(source_public_key)
47
48 base_fee = 100
49 # we are going to submit the transaction to the test network,
50 # so network_passphrase is `Network.TESTNET_NETWORK_PASSPHRASE`,
51 # if you want to submit to the public network, please use `Network.PUBLIC_NETWORK_
52 ↪PASSPHRASE`.
53 transaction = (
54     TransactionBuilder(
55         source_account=source_account,
56         network_passphrase=Network.TESTNET_NETWORK_PASSPHRASE,
57         base_fee=base_fee,
58     )
59     .add_text_memo("Hello, Stellar!") # Add a memo
60     # Add a payment operation to the transaction
61     # Send 350.1234567 XLM to receiver

```

(continues on next page)

(continued from previous page)

```

61     # Specify 350.1234567 lumens. Lumens are divisible to seven digits past the decimal.
62     .append_payment_op(receiver_public_key, Asset.native(), "350.1234567")
63     .set_timeout(30) # Make this transaction valid for the next 30 seconds only
64     .build()
65 )
66
67 # Sign this transaction with the secret key
68 # NOTE: signing is transaction is network specific. Test network transactions
69 # won't work in the public network. To switch networks, use the Network object
70 # as explained above (look for stellar_sdk.network.Network).
71 transaction.sign(source_keypair)
72
73 # Let's see the XDR (encoded in base64) of the transaction we just built
74 print(transaction.to_xdr())
75
76 # Submit the transaction to the Horizon server.
77 # The Horizon server will then submit the transaction into the network for us.
78 response = server.submit_transaction(transaction)
79 print(response)

```

1.7.2 Path Payment

In the example below we're sending 1000 XLM (at max) from *GABJLI6IVBKJ7HIC5NN7HHDCIEW3CMWQ2DWYHREQQUFWSWZ2* to *GBBM6BKZPEHWYO3E3YKREDPQXMS4VK35YLNU7NFBRI26RAN7GI5POFBB*. Destination Asset will be *GBP* issued by *GASOCNHNNLYFNMDJYQ3XFMI7BYHIOCFW3GJEOWRPEGK2TDPGTG2E5EDW*. Assets will be exchanged using the following path:

- *USD* issued by *GBBM6BKZPEHWYO3E3YKREDPQXMS4VK35YLNU7NFBRI26RAN7GI5POFBB*
- *EUR* issued by *GDTNXRLOJD2YEBPKK7KCMR7J33AAG5VZXHAJTHIG736D6LVEFLLLKPD*

The *path payment* will cause the destination address to get 5.5 GBP. It will cost the sender no more than 1000 XLM. In this example there will be 3 exchanges, XLM->USD, USD->EUR, EUR->GBP.

```

1  """
2  A path payment sends an amount of a specific asset to a destination account through a
3  ↪ path of offers.
4  Since the asset sent (e.g., 450 XLM) can be different from the asset received (e.g, 6
5  ↪ BTC),
6  path payments allow for the simultaneous transfer and conversion of currencies.
7
8  A Path Payment Strict Send allows a user to specify the amount of the asset to send.
9  The amount received will vary based on offers in the order books. If you would like to
10 instead specify the amount received, use the Path Payment Strict Receive operation.
11
12 See: https://developers.stellar.org/docs/start/list-of-operations/#path-payment-strict-
13 ↪ send
14 See: https://youtu.be/Kz1SgSPStz8
15 """
16 from stellar_sdk import Asset, Keypair, Network, Server, TransactionBuilder
17
18 server = Server(horizon_url="https://horizon-testnet.stellar.org")

```

(continues on next page)

(continued from previous page)

```

16 source_keypair = Keypair.from_secret(
17     "SA6XHAH4GNLRWWWF6TEVEWNS44CBNFAJWHWOPZCVZOUXSQA7BOYN7XHC"
18 )
19
20 source_account = server.load_account(account_id=source_keypair.public_key)
21
22 path = [
23     Asset("USD", "GBBM6BKZPEHWYO3E3YKREDPQXMS4VK35YLNUN7NFBRI26RAN7GI5POFBB"),
24     Asset("EUR", "GDTNXRLOJD2YEBPKK7KCMR7J33AAG5VZXHAJTHIG736D6LVEFLLKPD"),
25 ]
26 transaction = (
27     TransactionBuilder(
28         source_account=source_account,
29         network_passphrase=Network.TESTNET_NETWORK_PASSPHRASE,
30         base_fee=100,
31     )
32     .append_path_payment_strict_receive_op(
33         destination="GBBM6BKZPEHWYO3E3YKREDPQXMS4VK35YLNUN7NFBRI26RAN7GI5POFBB",
34         send_asset=Asset.native(),
35         send_max="1000",
36         dest_asset=Asset(
37             "GBP", "GASOCNHNLYFNMDJYQ3XFMI7BYHIOCFW3GJEOWRPEGK2TDPGTG2E5EDW"
38         ),
39         dest_amount="5.50",
40         path=path,
41     )
42     .set_timeout(30)
43     .build()
44 )
45 transaction.sign(source_keypair)
46 response = server.submit_transaction(transaction)

```

1.8 Asynchronous

Now we have supported the use of asynchronous methods to submit transactions, py-stellar-base gives you the choice, rather than forcing you into always writing async; sync code is easier to write, generally safer, and has many more libraries to choose from.

The following is an example of send a payment by an asynchronous method, the same example of using the synchronization method can be found [here](#):

```

1  """
2  The effect of this example is the same as `payment.py`, but this example is asynchronous.
3
4  Create, sign, and submit a transaction using Python Stellar SDK.
5
6  Assumes that you have the following items:
7  1. Secret key of a funded account to be the source account
8  2. Public key of an existing account as a recipient
9  These two keys can be created and funded by the friendbot at

```

(continues on next page)

(continued from previous page)

```

10     https://www.stellar.org/laboratory/ under the heading "Quick Start: Test Account"
11 3. Access to Python Stellar SDK (https://github.com/StellarCN/py-stellar-base) through
    ↪ Python shell.
12
13 See: https://developers.stellar.org/docs/start/list-of-operations/#payment
14 """
15 import asyncio
16
17 from stellar_sdk import (
18     AiohttpClient,
19     Asset,
20     Keypair,
21     Network,
22     ServerAsync,
23     TransactionBuilder,
24 )
25
26
27 def create_account():
28     """To make this script work, create an account on the testnet."""
29     import requests
30
31     from stellar_sdk import Keypair
32
33     keypair = Keypair.random()
34     url = "https://friendbot.stellar.org"
35     _response = requests.get(url, params={"addr": keypair.public_key})
36     # Check _response.json() in case something goes wrong
37     return keypair
38
39
40 # The source account is the account we will be signing and sending from.
41 example_keypair = create_account()
42 source_secret_key = example_keypair.secret
43
44 # Derive Keypair object and public key (that starts with a G) from the secret
45 source_keypair = Keypair.from_secret(source_secret_key)
46 source_public_key = source_keypair.public_key
47
48 # We just send lumen to ourselves in this simple example
49 receiver_public_key = example_keypair.public_key
50
51
52 async def main():
53     # Configure StellarSdk to talk to the horizon instance hosted by Stellar.org
54     # To use the live network, set the hostname to 'horizon.stellar.org'
55     # When we use the `with` syntax, it automatically releases the resources it occupies.
56     async with ServerAsync(
57         horizon_url="https://horizon-testnet.stellar.org", client=AiohttpClient()
58     ) as server:
59         # Transactions require a valid sequence number that is specific to this account.
60         # We can fetch the current sequence number for the source account from Horizon.

```

(continues on next page)

(continued from previous page)

```

61     source_account = await server.load_account(source_public_key)
62
63     base_fee = 100
64     # we are going to submit the transaction to the test network,
65     # so network_passphrase is `Network.TESTNET_NETWORK_PASSPHRASE`,
66     # if you want to submit to the public network, please use `Network.PUBLIC_
↳NETWORK_PASSPHRASE`.
67     transaction = (
68         TransactionBuilder(
69             source_account=source_account,
70             network_passphrase=Network.TESTNET_NETWORK_PASSPHRASE,
71             base_fee=base_fee,
72         )
73         .add_text_memo("Hello, Stellar!") # Add a memo
74         # Add a payment operation to the transaction
75         # Send 350.1234567 XLM to receiver
76         # Specify 350.1234567 lumens. Lumens are divisible to seven digits past the
↳decimal.
77         .append_payment_op(receiver_public_key, Asset.native(), "350.1234567")
78         .set_timeout(30) # Make this transaction valid for the next 30 seconds only
79         .build()
80     )
81
82     # Sign this transaction with the secret key
83     # NOTE: signing is transaction is network specific. Test network transactions
84     # won't work in the public network. To switch networks, use the Network object
85     # as explained above (look for stellar_sdk.network.Network).
86     transaction.sign(source_keypair)
87
88     # Let's see the XDR (encoded in base64) of the transaction we just built
89     print(transaction.to_xdr())
90
91     # Submit the transaction to the Horizon server.
92     # The Horizon server will then submit the transaction into the network for us.
93     response = await server.submit_transaction(transaction)
94     print(response)
95
96
97 if __name__ == "__main__":
98     loop = asyncio.get_event_loop()
99     loop.run_until_complete(main())
100    loop.close()
101    # asyncio.run(main()) # Python 3.7+

```

The following example helps you listen to multiple endpoints asynchronously.

```

1     """
2     See: https://stellar-sdk.readthedocs.io/en/latest/asynchronous.html
3     """
4     import asyncio
5
6     from stellar_sdk import AiohttpClient, ServerAsync

```

(continues on next page)

```
7
8 HORIZON_URL = "https://horizon.stellar.org"
9
10
11 async def payments():
12     async with ServerAsync(HORIZON_URL, AiohttpClient()) as server:
13         async for payment in server.payments().cursor(cursor="now").stream():
14             print(f"Payment: {payment}")
15
16
17 async def effects():
18     async with ServerAsync(HORIZON_URL, AiohttpClient()) as server:
19         async for effect in server.effects().cursor(cursor="now").stream():
20             print(f"Effect: {effect}")
21
22
23 async def operations():
24     async with ServerAsync(HORIZON_URL, AiohttpClient()) as server:
25         async for operation in server.operations().cursor(cursor="now").stream():
26             print(f"Operation: {operation}")
27
28
29 async def transactions():
30     async with ServerAsync(HORIZON_URL, AiohttpClient()) as server:
31         async for transaction in server.transactions().cursor(cursor="now").stream():
32             print(f"Transaction: {transaction}")
33
34
35 async def listen():
36     await asyncio.gather(payments(), effects(), operations(), transactions())
37
38
39 if __name__ == "__main__":
40     asyncio.run(listen())
```

1.9 Multi-signature account

Multi-signature accounts can be used to require that transactions require multiple public keys to sign before they are considered valid. This is done by first configuring your account's **threshold** levels. Each operation has a threshold level of either low, medium, or high. You give each threshold level a number between 1-255 in your account. Then, for each key in your account, you assign it a weight (1-255, setting a 0 weight deletes the key). Any transaction must be signed with enough keys to meet the threshold.

For example, let's say you set your threshold levels; low = 1, medium = 2, high = 3. You want to send a payment operation, which is a medium threshold operation. Your master key has weight 1. Additionally, you have a secondary key associated with your account which has a weight of 1. Now, the transaction you submit for this payment must include both signatures of your master key and secondary key since their combined weight is 2 which is enough to authorize the payment operation.

In this example, we will:

- Add a second signer to the account

- Set our account's masterkey weight and threshold levels
- Create a multi signature transaction that sends a payment

```

1  """
2  Stellar uses signatures as authorization. Transactions always need authorization
3  from at least one public key in order to be considered valid. Generally,
4  transactions only need authorization from the public key of the source account.
5
6  Transaction signatures are created by cryptographically signing the
7  transaction object contents with a secret key. Stellar currently uses the ed25519_
8  ↪signature
9  scheme, but there's also a mechanism for adding additional types of public/private key_
10 ↪schemes.
11 A transaction with an attached signature is considered to have authorization from that_
12 ↪public key.
13
14 In two cases, a transaction may need more than one signature. If the transaction has
15 operations that affect more than one account, it will need authorization from every_
16 ↪account
17 in question. A transaction will also need additional signatures if the account associated
18 with the transaction has multiple public keys.
19
20 See: https://developers.stellar.org/docs/glossary/multisig/
21 """
22
23 from stellar_sdk import Asset, Keypair, Network, Server, Signer, TransactionBuilder
24
25 server = Server(horizon_url="https://horizon-testnet.stellar.org")
26 root_keypair = Keypair.from_secret(
27     "SA6XHAH4GNLRWWWF6TEVEWNS44CBNFAJWHWOPZCVZOUXSQA7BOYN7XHC"
28 )
29 root_account = server.load_account(account_id=root_keypair.public_key)
30 secondary_keypair = Keypair.from_secret(
31     "SAMZUAAPLRUH62HH3XE7NVD6ZSMTWPWGM6DS4X47HLVRHEBKP4U2H5E7"
32 )
33
34 secondary_signer = Signer.ed25519_public_key(
35     account_id=secondary_keypair.public_key, weight=1
36 )
37
38 transaction = (
39     TransactionBuilder(
40         source_account=root_account,
41         network_passphrase=Network.TESTNET_NETWORK_PASSPHRASE,
42         base_fee=100,
43     )
44     .append_set_options_op(
45         master_weight=1, # set master key weight
46         low_threshold=1,
47         med_threshold=2, # a payment is medium threshold
48         high_threshold=2, # make sure to have enough weight to add up to the high_
49 ↪threshold!
50         signer=secondary_signer,
51     )
52     .set_timeout(30)

```

(continues on next page)

(continued from previous page)

```

46     .build()
47 )
48
49 # only need to sign with the root signer as the 2nd signer won't
50 # be added to the account till after this transaction completes
51 transaction.sign(root_keypair)
52 response = server.submit_transaction(transaction)
53 print(response)
54
55 # now create a payment with the account that has two signers
56 destination = "GBA5SMM5OYA00PL6R773MV703CCLUDVLCWHIVVL3W4XTD3DA5FJ4JSEZ"
57 transaction = (
58     TransactionBuilder(
59         source_account=root_account,
60         network_passphrase=Network.TESTNET_NETWORK_PASSPHRASE,
61         base_fee=100,
62     )
63     .append_payment_op(destination=destination, amount="2000", asset=Asset.native())
64     .set_timeout(30)
65     .build()
66 )
67
68 # now we need to sign the transaction with both the root and the secondary_keypair
69 transaction.sign(root_keypair)
70 transaction.sign(secondary_keypair)
71 response = server.submit_transaction(transaction)
72 print(response)

```

1.10 XDR

XDR, also known as External Data Representation, is used throughout the Stellar network and protocol. The ledger, transactions, results, history, and even the messages passed between computers running stellar-core are encoded using XDR.

`stellar_sdk.xdr` module provides a complete ability to build and parse XDR.

This example shows how to parse XDR string into an XDR object.

```

1  """
2  This example shows how to parse XDR string into an XDR object.
3  But please note that if you need to parse a transaction envelope,
4  please refer to `parse_transaction_envelope.py`
5  """
6  from stellar_sdk.xdr import TransactionResult
7
8  result_xdr = "AAAAAAAAAGQAAAAAAAAAAQAAAAAAAAADAAAAAAAAAAAAAAAAABAAAAAD/
9  ↪jlpBCTX53ogvts02Ryn5Gj06gx0qW3/3ARB+gOh/nAAAAADGRC/
10 ↪wAAAAAAAAAAU5VQwAAAAAAR74W04Rz02ryJo940i0FU50KHIVQisRnpe9FWrqvumQAAAAAEFWjwjgcksQkG4uAAAAAAAAAAAAAAAA
11 ↪"
12 transaction_result = TransactionResult.from_xdr(result_xdr)

```

(continues on next page)

(continued from previous page)

```
10 print(transaction_result.fee_charged)
11 print(transaction_result.result.code)
```


API DOCUMENTATION

Here you'll find detailed documentation on specific functions, classes, and methods.

2.1 API Documentation

2.1.1 Account

class `stellar_sdk.account.Account`(*account*, *sequence*, *raw_data=None*)

The *Account* object represents a single account on the Stellar network and its sequence number.

Account tracks the sequence number as it is used by *TransactionBuilder*.

Normally, you can get an *Account* instance through `stellar_sdk.server.Server.load_account()` or `stellar_sdk.server_async.ServerAsync.load_account()`.

An example:

```
from stellar_sdk import Keypair, Server

server = Server(horizon_url="https://horizon-testnet.stellar.org")
source = Keypair.from_secret(
    ↪ "SBFZCHU5645DOKRWYBXVOXY2ELGJKFRX6VGGPRYUWHQ7PMXXJNDZFMKD")
# `account` can also be a muxed account
source_account = server.load_account(account=source.public_key)
```

See [Accounts](#) for more information.

Parameters

- **account** (`Union[str, MuxedAccount]`) – Account Id of the account (ex. "GB3KJPLFUYN5VL6R3GU3EGCGVCKFDSD7BEDX42HWG5BWFKB3KQGJJRMA") or muxed account (ex. "MBZSQ3YZMZEWL5ZRCEQ5CCSOTXCFCMKDGFPP4IEQN2KN6LCHCLI46AAAAAAAAAAE2L2QE")
- **sequence** (`int`) – Current sequence number of the account.
- **raw_data** (`Optional[Dict[str, Any]]`) – Raw horizon response data.

increment_sequence_number()

Increments sequence number in this object by one.

Return type `None`

load_ed25519_public_key_signers()

Load ed25519 public key signers.

Return type `List[Ed25519PublicKeySigner]`

property universal_account_id: `str`

Get the universal account id, if *account* is ed25519 public key, it will return ed25519 public key (ex. "GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"), otherwise it will return muxed account (ex. "MAAAAAAAAAAJURAAB2X52XFQP6FBXLGT6LWOOWMEXWHEWBDVRZ7V5WH34Y22MPFBHUHY")

Return type `str`

2.1.2 Asset

class `stellar_sdk.asset.Asset`(*code*, *issuer=None*)

The *Asset* object, which represents an asset and its corresponding issuer on the Stellar network.

The following example shows how to create an *Asset* object:

```
from stellar_sdk import Asset

native_asset = Asset.native() # You can also create a native asset through Asset(
↳ "XLM").
credit_alphanum4_asset = Asset("USD",
↳ "GBSKJPM2FM602C6GVZNAUAMGXZ6I4QIUPMNWVDN2NZULPWTV3GI2SOX")
credit_alphanum12_asset = Asset("BANANA",
↳ "GA6VT2PDD73TNNRYLPJPJYAAI7EGKBATZ7V562S7XY7TJD4GNOXRG60S")
print(f"Asset type: {credit_alphanum4_asset.type}\n"
      f"Asset code: {credit_alphanum4_asset.code}\n"
      f"Asset issuer: {credit_alphanum4_asset.issuer}\n"
      f"Is native asset: {credit_alphanum4_asset.is_native()}")
```

For more information about the formats used for asset codes and how issuers work on Stellar's network, see Stellar's [guide on assets](#).

Parameters

- **code** (`str`) – The asset code, in the formats specified in [Stellar's guide on assets](#).
- **issuer** (`Optional[str]`) – The account ID of the issuer. Note if the currency is the native currency (XLM (Lumens)), no issuer is necessary.

Raises

AssetCodeInvalidError: if *code* is invalid.

AssetIssuerInvalidError: if *issuer* is not a valid ed25519 public key.

static `check_if_asset_code_is_valid`(*code*)

Check whether the *code* passed in by the user is a valid asset code, if not, an exception will be thrown.

Parameters *code* (`str`) – The asset code.

Raises *AssetCodeInvalidError*: if *code* is invalid.

Return type `None`

classmethod `from_xdr_object`(*cls*, *xdr_object*)

Create a *Asset* from an XDR *Asset/ChangeTrustAsset/TrustLineAsset* object.

Please note that this function only supports processing the following types of assets:

- `ASSET_TYPE_NATIVE`
- `ASSET_TYPE_CREDIT_ALPHANUM4`
- `ASSET_TYPE_CREDIT_ALPHANUM12`

Parameters `xdr_object` (`Union[Asset, ChangeTrustAsset, TrustLineAsset]`) – The XDR Asset/ChangeTrustAsset/TrustLineAsset object.

Return type `Asset`

Returns A new `Asset` object from the given XDR object.

guess_asset_type()

Return the type of the asset, Can be one of following types: `native`, `credit_alphanum4` or `credit_alphanum12`.

Return type `str`

Returns The type of the asset.

is_native()

Return True if the `Asset` object is the native asset.

Return type `bool`

Returns True if the asset object is native, False otherwise.

classmethod native(cls)

Returns an asset object for the native asset.

Return type `Asset`

Returns An asset object for the native asset.

to_change_trust_asset_xdr_object()

Returns the xdr object for this asset.

Return type `ChangeTrustAsset`

Returns XDR ChangeTrustAsset object

to_dict()

Generate a dict for this object's attributes.

Return type `dict`

Returns A dict representing an `Asset`

to_trust_line_asset_xdr_object()

Returns the xdr object for this asset.

Return type `TrustLineAsset`

Returns XDR TrustLineAsset object

to_xdr_object()

Returns the xdr object for this asset.

Return type `Asset`

Returns XDR Asset object

property type: `str`

Return the type of the asset, can be one of following types: `native`, `credit_alphanum4` or `credit_alphanum12`

Return type `str`

Returns The type of the asset.

2.1.3 Call Builder

AccountsCallBuilder

class `stellar_sdk.call_builder.call_builder_sync.AccountsCallBuilder`(*horizon_url, client*)

Creates a new *AccountsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.accounts()`.

See [List All Accounts](#) for more information.

Parameters

- **horizon_url** – Horizon server URL.
- **client** (*BaseSyncClient*) – The client instance used to send request.

account_id(*account_id*)

Returns information and links relating to a single account. The balances section in the returned JSON will also list all the trust lines this account has set up.

See [Retrieve an Account](#) for more information.

Parameters **account_id** (*str*) – account id, for example:
"GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current AccountCallBuilder instance

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_asset(*asset*)

Filtering accounts who have a trustline to an asset. The result is a list of accounts.

See [List All Accounts](#) for more information.

Parameters **asset** (*Asset*) – an issued asset

Returns current AccountCallBuilder instance

for_liquidity_pool(*liquidity_pool_id*)

Filtering accounts who have a trustline for the given pool. The result is a list of accounts.

See [List All Accounts](#) for more information.

Parameters **liquidity_pool_id** (`str`) – The ID of the liquidity pool in hex string., for example: "dd7b1ab831c273310ddb6c6f97870aa83c2fbd78ce22aded37ecbf4f3380fac7"

Returns current AccountCallBuilder instance

for_signer(*signer*)

Filtering accounts who have a given signer. The result is a list of accounts.

See [List All Accounts](#) for more information.

Parameters **signer** (`str`) – signer's account id, for example: "GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current AccountCallBuilder instance

for_sponsor(*sponsor*)

Filtering accounts where the given account is sponsoring the account or any of its sub-entries.

See [List All Accounts](#) for more information.

Parameters **sponsor** (`str`) – the sponsor id, for example: "GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current AccountCallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

AssetsCallBuilder

class stellar_sdk.call_builder.call_builder_sync.**AssetsCallBuilder**(*horizon_url, client*)

Creates a new *AssetsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.assets()`.

See [List All Assets](#) for more information.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseSyncClient*) – The client instance used to send request.

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFound: if `status_code == 404`

BadRequest: if `400 <= status_code < 500` and `status_code != 404`

BadResponse: if `500 <= status_code < 600`

UnknownRequest: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets `cursor` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current `CallBuilder` instance

for_code(*asset_code*)

This endpoint filters all assets by the asset code.

See [List All Assets](#) for more information.

Parameters **asset_code** (*str*) – asset code, for example: *USD*

Returns current `AssetCallBuilder` instance

for_issuer(*asset_issuer*)

This endpoint filters all assets by the asset issuer.

See [List All Assets](#) for more information.

Parameters **asset_issuer** (*str*) – asset issuer, for example:
"GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current `AssetCallBuilder` instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

See [Pagination](#)

Parameters `limit` (`int`) – Number of records the server should return.

Returns

`order(desc=True)`

Sets `order` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

Parameters `desc` (`bool`) – Sort direction, `True` to get desc sort direction, the default setting is `True`.

Returns current `CallBuilder` instance

`stream()`

Creates an `EventSource` that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an `EventSource`.

Raise `StreamClientError` - Failed to fetch stream resource.

ClaimableBalancesCallBuilder

`class stellar_sdk.call_builder.call_builder_sync.ClaimableBalancesCallBuilder(horizon_url, client)`

Creates a new `ClaimableBalancesCallBuilder` pointed to server defined by `horizon_url`. Do not create this object directly, use `stellar_sdk.Server.claimable_balance()`.

See [List Claimable Balances](#) for more information.

Parameters

- `horizon_url` – Horizon server URL.
- `client` (`BaseSyncClient`) – The client instance used to send request.

`call()`

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

`ConnectionError`: if you have not successfully connected to the server.

`NotFoundError`: if `status_code == 404`

`BadRequestError`: if `400 <= status_code < 500` and `status_code != 404`

`BadResponseError`: if `500 <= status_code < 600`

`UnknownRequestError`: if an unknown error occurs, please submit an issue

`claimable_balance(claimable_balance_id)`

Returns information and links relating to a single claimable balance.

See [List Claimable Balances](#) for more information.

Parameters `claimable_balance_id` (`str`) – claimable balance id

Returns current AccountCallBuilder instance

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_asset(*asset*)

Returns all claimable balances which provide a balance for the given asset.

See [List Claimable Balances](#) for more information.

Parameters **asset** (`Asset`) – an asset

Returns current ClaimableBalancesCallBuilder instance

for_claimant(*claimant*)

Returns all claimable balances which can be claimed by the given account ID.

See [List Claimable Balances](#) for more information.

Parameters **claimant** (`str`) – the account id, for example:
"GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current ClaimableBalancesCallBuilder instance

for_sponsor(*sponsor*)

Returns all claimable balances which are sponsored by the given account ID.

See [List Claimable Balances](#) for more information.

Parameters **sponsor** (`str`) – the sponsor id, for example:
"GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current ClaimableBalancesCallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

DataCallBuilder

class `stellar_sdk.call_builder.call_builder_sync.DataCallBuilder`(*horizon_url, client, account_id, data_name*)

Creates a new *DataCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.data()`.

See [Retrieve an Account's Data](#) for more information.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseSyncClient*) – The client instance used to send request.
- **account_id** (*str*) – account id, for example: "GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ"
- **data_name** (*str*) – Key name

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

- ConnectionError*: if you have not successfully connected to the server.
- NotFoundError*: if `status_code == 404`
- BadRequestError*: if `400 <= status_code < 500` and `status_code != 404`
- BadResponseError*: if `500 <= status_code < 600`
- UnknownRequestError*: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (*int*) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

EffectsCallBuilder

class `stellar_sdk.call_builder.call_builder_sync.EffectsCallBuilder`(*horizon_url, client*)

Creates a new *EffectsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.effects()`.

See [List All Effects](#) for more information.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseSyncClient*) – The client instance used to send request.

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_account(*account_id*)

This endpoint represents all effects that changed a given account. It will return relevant effects from the creation of the account to the current ledger.

See [Retrieve an Account's Effects](#) for more information.

Parameters **account_id** (**str**) – account id, for example:
"GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns this EffectCallBuilder instance

for_ledger(*sequence*)

Effects are the specific ways that the ledger was changed by any operation. This endpoint represents all effects that occurred in the given ledger.

See [Retrieve a Ledger's Effects](#) for more information.

Parameters **sequence** (**Union[int, str]**) – ledger sequence

Returns this EffectCallBuilder instance

for_liquidity_pool(*liquidity_pool_id*)

This endpoint represents all effects that occurred as a result of a given liquidity pool.

See [Liquidity Pools - Retrieve related Effects](#) for more information.

Parameters **liquidity_pool_id** (**str**) – The ID of the liquidity pool in hex string.

Returns this EffectsCallBuilder instance

for_operation(*operation_id*)

This endpoint represents all effects that occurred as a result of a given operation.

See [Retrieve an Operation's Effects](#) for more information.

Parameters **operation_id** (**Union[int, str]**) – operation ID

Returns this EffectCallBuilder instance

for_transaction(*transaction_hash*)

This endpoint represents all effects that occurred as a result of a given transaction.

See [Retrieve a Transaction's Effects](#) for more information.

Parameters **transaction_hash** (**str**) – transaction hash

Returns this EffectCallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (**int**) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (**bool**) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

FeeStatsCallBuilder

class `stellar_sdk.call_builder.call_builder_sync.FeeStatsCallBuilder`(*horizon_url, client*)

Creates a new *FeeStatsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.fee_stats()`.

See [Fee Stats](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (*BaseSyncClient*) – The client instance used to send request.

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

LedgersCallBuilder

class `stellar_sdk.call_builder.call_builder_sync.LedgersCallBuilder`(*horizon_url, client*)

Creates a new *LedgersCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.ledgers()`.

See [List All Ledgers](#) for more information.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseSyncClient*) – The client instance used to send request.

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFound: if `status_code == 404`

BadRequest: if `400 <= status_code < 500` and `status_code != 404`

BadResponse: if `500 <= status_code < 600`

UnknownRequest: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

ledger(*sequence*)

Provides information on a single ledger.

See [Retrieve a Ledger](#) for more information.

Parameters **sequence** (`Union[int, str]`) – Ledger sequence

Returns current LedgerCallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

LiquidityPoolsBuilder

class `stellar_sdk.call_builder.call_builder_sync.LiquidityPoolsBuilder`(*horizon_url, client*)

Creates a new `LiquidityPoolsBuilder` pointed to server defined by `horizon_url`. Do not create this object directly, use `stellar_sdk.Server.liquidity_pools()`.

See [List Liquidity Pools](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (`BaseSyncClient`) – The client instance used to send request.

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

`ConnectionError`: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets `cursor` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters `cursor` (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_account(*account_id*)

Filter pools for a specific account

See [List Liquidity Pools](#) for more information.

Parameters `account_id` (`str`) – account id

Returns current LiquidityPoolsBuilder instance

for_reserves(*reserves*)

Get pools by reserves.

Horizon will provide an endpoint to find all liquidity pools which contain a given set of reserve assets.

See [List Liquidity Pools](#) for more information.

Returns current LiquidityPoolsBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters `limit` (`int`) – Number of records the server should return.

Returns

liquidity_pool(*liquidity_pool_id*)

Provides information on a liquidity pool.

See [Retrieve a Liquidity Pool](#) for more information.

Parameters `liquidity_pool_id` (`str`) – The ID of the liquidity pool in hex string.

Returns current LiquidityPoolsBuilder instance

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters `desc` (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

OffersCallBuilder

class `stellar_sdk.call_builder.call_builder_sync.OffersCallBuilder`(*horizon_url, client*)

Creates a new *OffersCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.offers()`.

See [List All Offers](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (`BaseSyncClient`) – The client instance used to send request.

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_buying(*buying*)

Returns all offers buying an asset.

People on the Stellar network can make offers to buy or sell assets. This endpoint represents all the current offers, allowing filtering by *seller*, *selling_asset* or *buying_asset*.

See [List All Offers](#) for more information.

Parameters **buying** (`Asset`) – The asset being bought.

Returns this OffersCallBuilder instance

for_seller(*seller*)

Returns all offers where the given account is the seller.

People on the Stellar network can make offers to buy or sell assets. This endpoint represents all the current offers, allowing filtering by *seller*, *selling_asset* or *buying_asset*.

See [List All Offers](#) for more information.

Parameters *seller* (*str*) – Account ID of the offer creator

Returns this OffersCallBuilder instance

for_selling(*selling*)

Returns all offers selling an asset.

People on the Stellar network can make offers to buy or sell assets. This endpoint represents all the current offers, allowing filtering by *seller*, *selling_asset* or *buying_asset*.

See [List All Offers](#) for more information.

Parameters *selling* (*Asset*) – The asset being sold.

Returns this OffersCallBuilder instance

for_sponsor(*sponsor*)

Filtering offers where the given account is sponsoring the offer entry.

See [List All Offers](#) for more information.

Parameters *sponsor* (*str*) – the sponsor id, for example:
"GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current OffersCallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters *limit* (*int*) – Number of records the server should return.

Returns

offer(*offer_id*)

Returns information and links relating to a single offer.

See [Retrieve an Offer](#) for more information.

Parameters *offer_id* (*Union[str, int]*) – Offer ID.

Returns this OffersCallBuilder instance

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters *desc* (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type *Generator[Dict[str, Any], None, None]*

Returns an EventSource.

Raise StreamClientError - Failed to fetch stream resource.

OperationsCallBuilder

class stellar_sdk.call_builder.call_builder_sync.**OperationsCallBuilder**(*horizon_url, client*)

Creates a new *OperationsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.operations()`.

See [List All Operations](#) for more information.

Parameters

- **horizon_url** – Horizon server URL.
- **client** (*BaseSyncClient*) – The client instance used to send request.

call()

Triggers a HTTP request using this builder's current configuration.

Return type Dict[str, Any]

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (*Union[int, str]*) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_account(*account_id*)

This endpoint represents all operations that were included in valid transactions that affected a particular account.

See [Retrieve an Account's Operations](#) for more information.

Parameters **account_id** (*str*) – Account ID

Returns this OperationCallBuilder instance

for_claimable_balance(*claimable_balance_id*)

This endpoint represents successful operations referencing a given claimable balance and can be used in streaming mode.

See [Claimable Balances - Retrieve related Operations](#) for more information.

Parameters **claimable_balance_id** (*str*) – This claimable balance's id encoded in a hex string representation.

Returns this OperationCallBuilder instance

for_ledger(*sequence*)

This endpoint returns all operations that occurred in a given ledger.

See [Retrieve a Ledger's Operations](#) for more information.

Parameters **sequence** (`Union[int, str]`) – Sequence ID

Returns this OperationCallBuilder instance

for_liquidity_pool(*liquidity_pool_id*)

This endpoint represents all operations that are part of a given liquidity pool.

See [Liquidity Pools - Retrieve related Operations](#) for more information.

Parameters **liquidity_pool_id** (`str`) – The ID of the liquidity pool in hex string.

Returns this OperationCallBuilder instance

for_transaction(*transaction_hash*)

This endpoint represents all operations that are part of a given transaction.

See [Retrieve a Transaction's Operations](#) for more information.

Parameters **transaction_hash** (`str`) – Transaction Hash

Returns this OperationCallBuilder instance

include_failed(*include_failed*)

Adds a parameter defining whether to include failed transactions. By default only operations of successful transactions are returned.

Parameters **include_failed** (`bool`) – Set to *True* to include operations of failed transactions.

Returns current OperationsCallBuilder instance

join(*join*)

join represents *join* param in queries, currently only supports *transactions*

Parameters **join** (`str`) – join represents *join* param in queries, currently only supports *transactions*

Returns current OperationsCallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (`int`) – Number of records the server should return.

Returns

operation(*operation_id*)

The operation details endpoint provides information on a single operation. The operation ID provided in the `id` argument specifies which operation to load.

See [Retrieve an Operation](#) for more information.

Parameters **operation_id** (`Union[int, str]`) – Operation ID

Returns this OperationCallBuilder instance

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters *desc* (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

OrderbookCallBuilder

class `stellar_sdk.call_builder.call_builder_sync.OrderbookCallBuilder`(*horizon_url, client, selling, buying*)

Creates a new *OrderbookCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.orderbook()`.

See [Orderbook](#) for more information.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseSyncClient*) – The client instance used to send request.
- **selling** (*Asset*) – Asset being sold
- **buying** (*Asset*) – Asset being bought

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFound: if `status_code == 404`

BadRequest: if `400 <= status_code < 500` and `status_code != 404`

BadResponse: if `500 <= status_code < 600`

UnknownRequest: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters `cursor` (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters `limit` (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters `desc` (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

PaymentsCallBuilder

class `stellar_sdk.call_builder.call_builder_sync.PaymentsCallBuilder`(*horizon_url, client*)

Creates a new `PaymentsCallBuilder` pointed to server defined by `horizon_url`. Do not create this object directly, use `stellar_sdk.Server.payments()`.

See [List All Payments](#) for more information.

Parameters

- `horizon_url` (`str`) – Horizon server URL.
- `client` (`BaseSyncClient`) – The client instance used to send request.

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

`ConnectionError`: if you have not successfully connected to the server.

`NotFoundError`: if `status_code == 404`

`BadRequestError`: if `400 <= status_code < 500` and `status_code != 404`

BadRequestError: if $500 \leq \text{status_code} < 600$

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters *cursor* (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_account(*account_id*)

This endpoint responds with a collection of Payment operations where the given account was either the sender or receiver.

See [Retrieve an Account's Payments](#) for more information.

Parameters *account_id* (`str`) – Account ID

Returns current PaymentsCallBuilder instance

for_ledger(*sequence*)

This endpoint represents all payment operations that are part of a valid transactions in a given ledger.

See [Retrieve a Ledger's Payments](#) for more information.

Parameters *sequence* (`Union[int, str]`) – Ledger sequence

Returns current PaymentsCallBuilder instance

for_transaction(*transaction_hash*)

This endpoint represents all payment operations that are part of a given transaction.

P.S. The documentation provided by SDF seems to be missing this API.

Parameters *transaction_hash* (`str`) – Transaction hash

Returns current PaymentsCallBuilder instance

include_failed(*include_failed*)

Adds a parameter defining whether to include failed transactions. By default only payments of successful transactions are returned.

Parameters *include_failed* (`bool`) – Set to True to include payments of failed transactions.

Returns current PaymentsCallBuilder instance

join(*join*)

join represents *join* param in queries, currently only supports *transactions*

Parameters *join* (`str`) – *join* represents *join* param in queries, currently only supports *transactions*

Returns current OperationsCallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters *limit* (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters *desc* (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

RootCallBuilder

class `stellar_sdk.call_builder.call_builder_sync.RootCallBuilder`(*horizon_url, client*)

Creates a new *RootCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.root()`.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseSyncClient*) – The client instance used to send request.

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters *cursor* (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (*int*) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

StrictReceivePathsCallBuilder

```
class stellar_sdk.call_builder.call_builder_sync.StrictReceivePathsCallBuilder(horizon_url,  
                                                                           client,  
                                                                           source, desti-  
                                                                           nation_asset,  
                                                                           destina-  
                                                                           tion_amount)
```

Creates a new *StrictReceivePathsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.strict_receive_paths()`.

The Stellar Network allows payments to be made across assets through path payments. A path payment specifies a series of assets to route a payment through, from source asset (the asset debited from the payer) to destination asset (the asset credited to the payee).

A path search is specified using:

- The source address or source assets.
- The asset and amount that the destination account should receive.

As part of the search, horizon will load a list of assets available to the source address and will find any payment paths from those source assets to the desired destination asset. The search's amount parameter will be used to determine if there a given path can satisfy a payment of the desired amount.

If a list of assets is passed as the source, horizon will find any payment paths from those source assets to the desired destination asset.

See [List Strict Receive Payment Paths](#) for more information.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseSyncClient*) – The client instance used to send request.
- **source** (*Union[str, List[Asset]]*) – The sender’s account ID or a list of Assets. Any returned path must use a source that the sender can hold.
- **destination_asset** (*Asset*) – The destination asset.
- **destination_amount** (*str*) – The amount, denominated in the destination asset, that any returned path should be able to satisfy.

call()

Triggers a HTTP request using this builder’s current configuration.

Return type *Dict[str, Any]*

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFound: if `status_code == 404`

BadRequest: if `400 <= status_code < 500` and `status_code != 404`

BadResponse: if `500 <= status_code < 600`

UnknownRequest: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (*Union[int, str]*) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (*int*) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an `EventSource`.

Raise `StreamClientError` - Failed to fetch stream resource.

StrictSendPathsCallBuilder

```
class stellar_sdk.call_builder.call_builder_sync.StrictSendPathsCallBuilder(horizon_url,  
                                                                           client,  
                                                                           source_asset,  
                                                                           source_amount,  
                                                                           destination)
```

Creates a new `StrictSendPathsCallBuilder` pointed to server defined by `horizon_url`. Do not create this object directly, use `stellar_sdk.Server.strict_send_paths()`.

The Stellar Network allows payments to be made across assets through path payments. A strict send path payment specifies a series of assets to route a payment through, from source asset (the asset debited from the payer) to destination asset (the asset credited to the payee).

A strict send path search is specified using:

- The source asset
- The source amount
- The destination assets or destination account.

As part of the search, horizon will load a list of assets available to the source address and will find any payment paths from those source assets to the desired destination asset. The search's `source_amount` parameter will be used to determine if there a given path can satisfy a payment of the desired amount.

See [List Strict Send Payment Paths](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (`BaseSyncClient`) – The client instance used to send request.
- **source_asset** (`Asset`) – The asset to be sent.
- **source_amount** (`str`) – The amount, denominated in the source asset, that any returned path should be able to satisfy.
- **destination** (`Union[str, List[Asset]]`) – The destination account or the destination assets.

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

`ConnectionError`: if you have not successfully connected to the server.

`NotFoundError`: if `status_code == 404`

`BadRequestError`: if `400 <= status_code < 500` and `status_code != 404`

`BadResponseError`: if `500 <= status_code < 600`

`UnknownRequestError`: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

TradeAggregationsCallBuilder

```
class stellar_sdk.call_builder.call_builder_sync.TradeAggregationsCallBuilder(horizon_url,
                                                                           client, base,
                                                                           counter,
                                                                           resolution,
                                                                           start_time=None,
                                                                           end_time=None,
                                                                           offset=None)
```

Creates a new `TradeAggregationsCallBuilder` pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.trade_aggregations()`.

Trade Aggregations facilitate efficient gathering of historical trade data.

See [List Trade Aggregations](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (`BaseSyncClient`) – The client instance used to send request.

- **base** (*Asset*) – base asset
- **counter** (*Asset*) – counter asset
- **resolution** (*int*) – segment duration as millis since epoch. *Supported values are 1 minute (60000), 5 minutes (300000), 15 minutes (900000), 1 hour (3600000), 1 day (86400000) and 1 week (604800000).*
- **start_time** (*Optional[int]*) – lower time boundary represented as millis since epoch
- **end_time** (*Optional[int]*) – upper time boundary represented as millis since epoch
- **offset** (*Optional[int]*) – segments can be offset using this parameter. Expressed in milliseconds. *Can only be used if the resolution is greater than 1 hour. Value must be in whole hours, less than the provided resolution, and less than 24 hours.*

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets `cursor` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

See [Pagination](#)

Parameters `cursor` (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current `CallBuilder` instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

See [Pagination](#)

Parameters `limit` (*int*) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

Parameters `desc` (*bool*) – Sort direction, `True` to get desc sort direction, the default setting is `True`.

Returns current `CallBuilder` instance

stream()

Creates an `EventSource` that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an `EventSource`.

Raise `StreamClientError` - Failed to fetch stream resource.

TradesCallBuilder

class `stellar_sdk.call_builder.call_builder_sync.TradesCallBuilder`(*horizon_url, client*)

Creates a new *TradesCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.trades()`.

See [List All Trades](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (*BaseSyncClient*) – The client instance used to send request.

call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

- ConnectionError*: if you have not successfully connected to the server.
- NotFoundError*: if `status_code == 404`
- BadRequestError*: if `400 <= status_code < 500` and `status_code != 404`
- BadResponseError*: if `500 <= status_code < 600`
- UnknownRequestError*: if an unknown error occurs, please submit an issue

cursor(cursor)

Sets `cursor` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

See [Pagination](#)

Parameters `cursor` (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current `CallBuilder` instance

for_account(account_id)

Filter trades for a specific account

See [Retrieve an Account's Trades](#) for more information.

Parameters `account_id` (`str`) – account id

Returns current `TradesCallBuilder` instance

for_asset_pair(base, counter)

Filter trades for a specific asset pair (orderbook)

See [List All Trades](#) for more information.

Parameters

- **base** (*Asset*) – base asset
- **counter** (*Asset*) – counter asset

Returns current TradesCallBuilder instance

for_liquidity_pool(*liquidity_pool_id*)

Filter trades for a specific liquidity pool.

See [Liquidity Pools - Retrieve related Trades](#)

Parameters **liquidity_pool_id** (*str*) – The ID of the liquidity pool in hex string.

Returns current TradesCallBuilder instance

for_offer(*offer_id*)

Filter trades for a specific offer

See [List All Trades](#) for more information.

Parameters **offer_id** (*Union[int, str]*) – offer id

Returns current TradesCallBuilder instance

for_trade_type(*trade_type*)

Filter trades for a specific trade type

Horizon will reject requests which attempt to set *trade_type* to `liquidity_pools` when using the offer id filter.

Parameters **trade_type** (*str*) – trade type, the currently supported types are "orderbook", "liquidity_pools" and "all", defaults to "all".

Returns current TradesCallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (*int*) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

TransactionsCallBuilder

class `stellar_sdk.call_builder.call_builder_sync.TransactionsCallBuilder`(*horizon_url, client*)
Creates a new *TransactionsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.Server.transactions()`.

See [List All Transactions](#) for more information.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseSyncClient*) – The client instance used to send request.

call()

Triggers a HTTP request using this builder’s current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

- ConnectionError*: if you have not successfully connected to the server.
- NotFound**Error*: if `status_code == 404`
- BadRequest**Error*: if `400 <= status_code < 500` and `status_code != 404`
- BadResponse**Error*: if `500 <= status_code < 600`
- UnknownRequest**Error*: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current `CallBuilder` instance

for_account(*account_id*)

This endpoint represents all transactions that affected a given account.

See [Retrieve an Account’s Transactions](#) for more information.

Parameters **account_id** (*str*) – account id

Returns current `TransactionsCallBuilder` instance

for_claimable_balance(*claimable_balance_id*)

This endpoint represents all transactions referencing a given claimable balance and can be used in streaming mode.

See [Claimable Balances - Retrieve related Transactions](#)

Parameters **claimable_balance_id** (*str*) – This claimable balance’s id encoded in a hex string representation.

Returns current `TransactionsCallBuilder` instance

for_ledger(*sequence*)

This endpoint represents all transactions in a given ledger.

See [Retrieve a Ledger’s Transactions](#) for more information.

Parameters `sequence` (`Union[str, int]`) – ledger sequence

Returns current `TransactionsCallBuilder` instance

for_liquidity_pool(*liquidity_pool_id*)

This endpoint represents all transactions referencing a given liquidity pool.

See [Liquidity Pools - Retrieve related Transactions](#)

Parameters `liquidity_pool_id` (`str`) – The ID of the liquidity pool in hex string.

Returns this `TransactionsCallBuilder` instance

include_failed(*include_failed*)

Adds a parameter defining whether to include failed transactions. By default only transactions of successful transactions are returned.

Parameters `include_failed` (`bool`) – Set to `True` to include failed transactions.

Returns current `TransactionsCallBuilder` instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

See [Pagination](#)

Parameters `limit` (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

Parameters `desc` (`bool`) – Sort direction, `True` to get desc sort direction, the default setting is `True`.

Returns current `CallBuilder` instance

stream()

Creates an `EventSource` that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `Generator[Dict[str, Any], None, None]`

Returns an `EventSource`.

Raise `StreamClientError` - Failed to fetch stream resource.

transaction(*transaction_hash*)

The transaction details endpoint provides information on a single transaction. The transaction hash provided in the hash argument specifies which transaction to load.

See [Retrieve a Transaction](#) for more information.

Parameters `transaction_hash` (`str`) – transaction hash

Returns current `TransactionsCallBuilder` instance

2.1.4 Call Builder Async

AccountsCallBuilder

class `stellar_sdk.call_builder.call_builder_async.AccountsCallBuilder`(*horizon_url*, *client*)
 Creates a new *AccountsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.accounts()`.

See [List All Accounts](#) for more information.

Parameters

- **horizon_url** – Horizon server URL.
- **client** (*BaseAsyncClient*) – The client instance used to send request.

account_id(*account_id*)

Returns information and links relating to a single account. The balances section in the returned JSON will also list all the trust lines this account has set up.

See [Retrieve an Account](#) for more information.

Parameters **account_id** (*str*) – account id, for example:
 "GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current AccountCallBuilder instance

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.
NotFoundError: if `status_code == 404`
BadRequestError: if `400 <= status_code < 500` and `status_code != 404`
BadResponseError: if `500 <= status_code < 600`
UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_asset(*asset*)

Filtering accounts who have a trustline to an asset. The result is a list of accounts.

See [List All Accounts](#) for more information.

Parameters **asset** (*Asset*) – an issued asset

Returns current AccountCallBuilder instance

for_liquidity_pool(*liquidity_pool_id*)

Filtering accounts who have a trustline for the given pool. The result is a list of accounts.

See [List All Accounts](#) for more information.

Parameters **liquidity_pool_id** (*str*) – The ID of the liquidity pool in hex string., for example: "dd7b1ab831c273310ddbec6f97870aa83c2fbd78ce22aded37ecbf4f3380fac7"

Returns current AccountCallBuilder instance

for_signer(*signer*)

Filtering accounts who have a given signer. The result is a list of accounts.

See [List All Accounts](#) for more information.

Parameters **signer** (*str*) – signer's account id, for example: "GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current AccountCallBuilder instance

for_sponsor(*sponsor*)

Filtering accounts where the given account is sponsoring the account or any of its sub-entries.

See [List All Accounts](#) for more information.

Parameters **sponsor** (*str*) – the sponsor id, for example: "GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current AccountCallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (*int*) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

AssetsCallBuilder

class stellar_sdk.call_builder.call_builder_async.**AssetsCallBuilder**(*horizon_url*, *client*)

Creates a new *AssetsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.assets()`.

See [List All Assets](#) for more information.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseAsyncClient*) – The client instance used to send request.

async `call()`

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

- ConnectionError*: if you have not successfully connected to the server.
- NotFoundError*: if `status_code == 404`
- BadRequestError*: if `400 <= status_code < 500` and `status_code != 404`
- BadResponseError*: if `500 <= status_code < 600`
- UnknownRequestError*: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current `CallBuilder` instance

for_code(*asset_code*)

This endpoint filters all assets by the asset code.

See [List All Assets](#) for more information.

Parameters **asset_code** (*str*) – asset code, for example: *USD*

Returns current `AssetCallBuilder` instance

for_issuer(*asset_issuer*)

This endpoint filters all assets by the asset issuer.

See [List All Assets](#) for more information.

Parameters **asset_issuer** (*str*) – asset issuer, for example:
"GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current `AssetCallBuilder` instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

See [Pagination](#)

Parameters `limit` (`int`) – Number of records the server should return.

Returns

`order(desc=True)`

Sets `order` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

Parameters `desc` (`bool`) – Sort direction, `True` to get desc sort direction, the default setting is `True`.

Returns current `CallBuilder` instance

`stream()`

Creates an `EventSource` that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an `EventSource`.

Raise `StreamClientError` - Failed to fetch stream resource.

ClaimableBalancesCallBuilder

`class stellar_sdk.call_builder.call_builder_async.ClaimableBalancesCallBuilder` (`horizon_url`, `client`)

Creates a new `ClaimableBalancesCallBuilder` pointed to server defined by `horizon_url`. Do not create this object directly, use `stellar_sdk.ServerAsync.claimable_balance()`.

See [List Claimable Balances](#) for more information.

Parameters

- `horizon_url` – Horizon server URL.
- `client` (`BaseAsyncClient`) – The client instance used to send request.

`async call()`

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

`ConnectionError`: if you have not successfully connected to the server.

`NotFoundError`: if `status_code == 404`

`BadRequestError`: if `400 <= status_code < 500` and `status_code != 404`

`BadResponseError`: if `500 <= status_code < 600`

`UnknownRequestError`: if an unknown error occurs, please submit an issue

`claimable_balance(claimable_balance_id)`

Returns information and links relating to a single claimable balance.

See [List Claimable Balances](#) for more information.

Parameters `claimable_balance_id` (`str`) – claimable balance id

Returns current AccountCallBuilder instance

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_asset(*asset*)

Returns all claimable balances which provide a balance for the given asset.

See [List Claimable Balances](#) for more information.

Parameters **asset** (`Asset`) – an asset

Returns current ClaimableBalancesCallBuilder instance

for_claimant(*claimant*)

Returns all claimable balances which can be claimed by the given account ID.

See [List Claimable Balances](#) for more information.

Parameters **claimant** (`str`) – the account id, for example:
"GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current ClaimableBalancesCallBuilder instance

for_sponsor(*sponsor*)

Returns all claimable balances which are sponsored by the given account ID.

See [List Claimable Balances](#) for more information.

Parameters **sponsor** (`str`) – the sponsor id, for example:
"GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current ClaimableBalancesCallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

DataCallBuilder

class `stellar_sdk.call_builder.call_builder_async.DataCallBuilder`(*horizon_url, client, account_id, data_name*)

Creates a new *DataCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.data()`.

See [Retrieve an Account's Data](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (`BaseAsyncClient`) – The client instance used to send request.
- **account_id** (`str`) – account id, for example: "GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ"
- **data_name** (`str`) – Key name

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

- ConnectionError*: if you have not successfully connected to the server.
- NotFoundError*: if `status_code == 404`
- BadRequestError*: if `400 <= status_code < 500` and `status_code != 404`
- BadResponseError*: if `500 <= status_code < 600`
- UnknownRequestError*: if an unknown error occurs, please submit an issue

cursor(cursor)

Sets `cursor` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters `cursor` (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit(limit)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters `limit` (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

EffectsCallBuilder

class `stellar_sdk.call_builder.call_builder_async.EffectsCallBuilder`(*horizon_url, client*)

Creates a new *EffectsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.effects()`.

See [List All Effects](#) for more information.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseAsyncClient*) – The client instance used to send request.

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_account(*account_id*)

This endpoint represents all effects that changed a given account. It will return relevant effects from the creation of the account to the current ledger.

See [Retrieve an Account's Effects](#) for more information.

Parameters **account_id** (**str**) – account id, for example:
"GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns this EffectCallBuilder instance

for_ledger(*sequence*)

Effects are the specific ways that the ledger was changed by any operation. This endpoint represents all effects that occurred in the given ledger.

See [Retrieve a Ledger's Effects](#) for more information.

Parameters **sequence** (**Union[int, str]**) – ledger sequence

Returns this EffectCallBuilder instance

for_liquidity_pool(*liquidity_pool_id*)

This endpoint represents all effects that occurred as a result of a given liquidity pool.

See [Liquidity Pools - Retrieve related Effects](#) for more information.

Parameters **liquidity_pool_id** (**str**) – The ID of the liquidity pool in hex string.

Returns this EffectsCallBuilder instance

for_operation(*operation_id*)

This endpoint represents all effects that occurred as a result of a given operation.

See [Retrieve an Operation's Effects](#) for more information.

Parameters **operation_id** (**Union[int, str]**) – operation ID

Returns this EffectCallBuilder instance

for_transaction(*transaction_hash*)

This endpoint represents all effects that occurred as a result of a given transaction.

See [Retrieve a Transaction's Effects](#) for more information.

Parameters **transaction_hash** (**str**) – transaction hash

Returns this EffectCallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (**int**) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (**bool**) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

FeeStatsCallBuilder

class `stellar_sdk.call_builder.call_builder_async.FeeStatsCallBuilder`(*horizon_url, client*)

Creates a new *FeeStatsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.fee_stats()`.

See [Fee Stats](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (`BaseAsyncClient`) – The client instance used to send request.

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters *desc* (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

LedgersCallBuilder

class `stellar_sdk.call_builder.call_builder_async.LedgersCallBuilder`(*horizon_url, client*)

Creates a new *LedgersCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.ledgers()`.

See [List All Ledgers](#) for more information.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseAsyncClient*) – The client instance used to send request.

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters *cursor* (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

ledger(*sequence*)

Provides information on a single ledger.

See [Retrieve a Ledger](#) for more information.

Parameters **sequence** (`Union[int, str]`) – Ledger sequence

Returns current LedgerCallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

LiquidityPoolsBuilder

class `stellar_sdk.call_builder.call_builder_async.LiquidityPoolsBuilder`(*horizon_url, client*)

Creates a new `LiquidityPoolsBuilder` pointed to server defined by `horizon_url`. Do not create this object directly, use `stellar_sdk.ServerAsync.liquidity_pools()`.

See [List Liquidity Pools](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (`BaseAsyncClient`) – The client instance used to send request.

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

`ConnectionError`: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets `cursor` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters `cursor` (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_account(*account_id*)

Filter pools for a specific account

See [List Liquidity Pools](#) for more information.

Parameters `account_id` (`str`) – account id

Returns current LiquidityPoolsBuilder instance

for_reserves(*reserves*)

Get pools by reserves.

Horizon will provide an endpoint to find all liquidity pools which contain a given set of reserve assets.

See [List Liquidity Pools](#) for more information.

Returns current LiquidityPoolsBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters `limit` (`int`) – Number of records the server should return.

Returns

liquidity_pool(*liquidity_pool_id*)

Provides information on a liquidity pool.

See [Retrieve a Liquidity Pool](#) for more information.

Parameters `liquidity_pool_id` (`str`) – The ID of the liquidity pool in hex string.

Returns current LiquidityPoolsBuilder instance

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters `desc` (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

OffersCallBuilder

class `stellar_sdk.call_builder.call_builder_async.OffersCallBuilder`(*horizon_url*, *client*)

Creates a new *OffersCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.offers()`.

See [List All Offers](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (`BaseAsyncClient`) – The client instance used to send request.

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(cursor)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_buying(buying)

Returns all offers buying an asset.

People on the Stellar network can make offers to buy or sell assets. This endpoint represents all the current offers, allowing filtering by *seller*, *selling_asset* or *buying_asset*.

See [List All Offers](#) for more information.

Parameters **buying** (`Asset`) – The asset being bought.

Returns this OffersCallBuilder instance

for_seller(seller)

Returns all offers where the given account is the seller.

People on the Stellar network can make offers to buy or sell assets. This endpoint represents all the current offers, allowing filtering by *seller*, *selling_asset* or *buying_asset*.

See [List All Offers](#) for more information.

Parameters `seller` (`str`) – Account ID of the offer creator

Returns this OffersCallBuilder instance

for_selling(*selling*)

Returns all offers selling an asset.

People on the Stellar network can make offers to buy or sell assets. This endpoint represents all the current offers, allowing filtering by *seller*, *selling_asset* or *buying_asset*.

See [List All Offers](#) for more information.

Parameters `selling` (`Asset`) – The asset being sold.

Returns this OffersCallBuilder instance

for_sponsor(*sponsor*)

Filtering offers where the given account is sponsoring the offer entry.

See [List All Offers](#) for more information.

Parameters `sponsor` (`str`) – the sponsor id, for example:
 "GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"

Returns current OffersCallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters `limit` (`int`) – Number of records the server should return.

Returns

offer(*offer_id*)

Returns information and links relating to a single offer.

See [Retrieve an Offer](#) for more information.

Parameters `offer_id` (`Union[str, int]`) – Offer ID.

Returns this OffersCallBuilder instance

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters `desc` (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise StreamClientError - Failed to fetch stream resource.

OperationsCallBuilder

class stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder(*horizon_url*, *client*)

Creates a new *OperationsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.operations()`.

See [List All Operations](#) for more information.

Parameters

- **horizon_url** – Horizon server URL.
- **client** (*BaseAsyncClient*) – The client instance used to send request.

async call()

Triggers a HTTP request using this builder's current configuration.

Return type Dict[str, Any]

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (Union[int, str]) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_account(*account_id*)

This endpoint represents all operations that were included in valid transactions that affected a particular account.

See [Retrieve an Account's Operations](#) for more information.

Parameters **account_id** (str) – Account ID

Returns this OperationCallBuilder instance

for_claimable_balance(*claimable_balance_id*)

This endpoint represents successful operations referencing a given claimable balance and can be used in streaming mode.

See [Claimable Balances - Retrieve related Operations](#) for more information.

Parameters **claimable_balance_id** (str) – This claimable balance's id encoded in a hex string representation.

Returns this `OperationCallBuilder` instance

for_ledger(*sequence*)

This endpoint returns all operations that occurred in a given ledger.

See [Retrieve a Ledger's Operations](#) for more information.

Parameters `sequence` (`Union[int, str]`) – Sequence ID

Returns this `OperationCallBuilder` instance

for_liquidity_pool(*liquidity_pool_id*)

This endpoint represents all operations that are part of a given liquidity pool.

See [Liquidity Pools - Retrieve related Operations](#) for more information.

Parameters `liquidity_pool_id` (`str`) – The ID of the liquidity pool in hex string.

Returns this `OperationCallBuilder` instance

for_transaction(*transaction_hash*)

This endpoint represents all operations that are part of a given transaction.

See [Retrieve a Transaction's Operations](#) for more information.

Parameters `transaction_hash` (`str`) – Transaction Hash

Returns this `OperationCallBuilder` instance

include_failed(*include_failed*)

Adds a parameter defining whether to include failed transactions. By default only operations of successful transactions are returned.

Parameters `include_failed` (`bool`) – Set to `True` to include operations of failed transactions.

Returns current `OperationsCallBuilder` instance

join(*join*)

`join` represents `join` param in queries, currently only supports `transactions`

Parameters `join` (`str`) – `join` represents `join` param in queries, currently only supports `transactions`

Returns current `OperationsCallBuilder` instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

See [Pagination](#)

Parameters `limit` (`int`) – Number of records the server should return.

Returns

operation(*operation_id*)

The operation details endpoint provides information on a single operation. The operation ID provided in the `id` argument specifies which operation to load.

See [Retrieve an Operation](#) for more information.

Parameters `operation_id` (`Union[int, str]`) – Operation ID

Returns this `OperationCallBuilder` instance

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters *desc* (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

OrderbookCallBuilder

class `stellar_sdk.call_builder.call_builder_async.OrderbookCallBuilder`(*horizon_url, client, selling, buying*)

Creates a new *OrderbookCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.orderbook()`.

See [Orderbook](#) for more information.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseAsyncClient*) – The client instance used to send request.
- **selling** (*Asset*) – Asset being sold
- **buying** (*Asset*) – Asset being bought

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters `cursor` (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters `limit` (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters `desc` (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

PaymentsCallBuilder

class `stellar_sdk.call_builder.call_builder_async.PaymentsCallBuilder`(*horizon_url, client*)

Creates a new `PaymentsCallBuilder` pointed to server defined by `horizon_url`. Do not create this object directly, use `stellar_sdk.ServerAsync.payments()`.

See [List All Payments](#) for more information.

Parameters

- `horizon_url` (`str`) – Horizon server URL.
- `client` (`BaseAsyncClient`) – The client instance used to send request.

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

`ConnectionError`: if you have not successfully connected to the server.

`NotFoundError`: if `status_code == 404`

`BadRequestError`: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if $500 \leq \text{status_code} < 600$

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters *cursor* (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_account(*account_id*)

This endpoint responds with a collection of Payment operations where the given account was either the sender or receiver.

See [Retrieve an Account's Payments](#) for more information.

Parameters *account_id* (`str`) – Account ID

Returns current PaymentsCallBuilder instance

for_ledger(*sequence*)

This endpoint represents all payment operations that are part of a valid transactions in a given ledger.

See [Retrieve a Ledger's Payments](#) for more information.

Parameters *sequence* (`Union[int, str]`) – Ledger sequence

Returns current PaymentsCallBuilder instance

for_transaction(*transaction_hash*)

This endpoint represents all payment operations that are part of a given transaction.

P.S. The documentation provided by SDF seems to be missing this API.

Parameters *transaction_hash* (`str`) – Transaction hash

Returns current PaymentsCallBuilder instance

include_failed(*include_failed*)

Adds a parameter defining whether to include failed transactions. By default only payments of successful transactions are returned.

Parameters *include_failed* (`bool`) – Set to True to include payments of failed transactions.

Returns current PaymentsCallBuilder instance

join(*join*)

join represents *join* param in queries, currently only supports *transactions*

Parameters *join* (`str`) – *join* represents *join* param in queries, currently only supports *transactions*

Returns current OperationsCallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters *limit* (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters *desc* (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

RootCallBuilder

class `stellar_sdk.call_builder.call_builder_async.RootCallBuilder`(*horizon_url, client*)

Creates a new *RootCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.root()`.

Parameters

- **horizon_url** (*str*) – Horizon server URL.
- **client** (*BaseAsyncClient*) – The client instance used to send request.

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters *cursor* (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

See [Pagination](#)

Parameters `limit` (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

Parameters `desc` (`bool`) – Sort direction, `True` to get desc sort direction, the default setting is `True`.

Returns current `CallBuilder` instance

stream()

Creates an `EventSource` that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an `EventSource`.

Raise `StreamClientError` - Failed to fetch stream resource.

StrictReceivePathsCallBuilder

```
class stellar_sdk.call_builder.call_builder_async.StrictReceivePathsCallBuilder(horizon_url,
                                                                              client,
                                                                              source,
                                                                              destina-
                                                                              tion_asset,
                                                                              destina-
                                                                              tion_amount)
```

Creates a new `StrictReceivePathsCallBuilder` pointed to server defined by `horizon_url`. Do not create this object directly, use `stellar_sdk.ServerAsync.strict_receive_paths()`.

The Stellar Network allows payments to be made across assets through path payments. A path payment specifies a series of assets to route a payment through, from source asset (the asset debited from the payer) to destination asset (the asset credited to the payee).

A path search is specified using:

- The source address or source assets.
- The asset and amount that the destination account should receive.

As part of the search, horizon will load a list of assets available to the source address and will find any payment paths from those source assets to the desired destination asset. The search's amount parameter will be used to determine if there a given path can satisfy a payment of the desired amount.

If a list of assets is passed as the source, horizon will find any payment paths from those source assets to the desired destination asset.

See [List Strict Receive Payment Paths](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (`BaseAsyncClient`) – The client instance used to send request.
- **source** (`Union[str, List[Asset]]`) – The sender’s account ID or a list of Assets. Any returned path must use a source that the sender can hold.
- **destination_asset** (`Asset`) – The destination asset.
- **destination_amount** (`str`) – The amount, denominated in the destination asset, that any returned path should be able to satisfy.

async call()

Triggers a HTTP request using this builder’s current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(cursor)

Sets `cursor` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit(limit)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (`int`) – Number of records the server should return.

Returns

order(desc=True)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

StrictSendPathsCallBuilder

```
class stellar_sdk.call_builder.call_builder_async.StrictSendPathsCallBuilder(horizon_url,
                                                                            client,
                                                                            source_asset,
                                                                            source_amount,
                                                                            destination)
```

Creates a new `StrictSendPathsCallBuilder` pointed to server defined by `horizon_url`. Do not create this object directly, use `stellar_sdk.ServerAsync.strict_send_paths()`.

The Stellar Network allows payments to be made across assets through path payments. A strict send path payment specifies a series of assets to route a payment through, from source asset (the asset debited from the payer) to destination asset (the asset credited to the payee).

A strict send path search is specified using:

- The source asset
- The source amount
- The destination assets or destination account.

As part of the search, horizon will load a list of assets available to the source address and will find any payment paths from those source assets to the desired destination asset. The search's `source_amount` parameter will be used to determine if there a given path can satisfy a payment of the desired amount.

See [List Strict Send Payment Paths](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (`BaseAsyncClient`) – The client instance used to send request.
- **source_asset** (`Asset`) – The asset to be sent.
- **source_amount** (`str`) – The amount, denominated in the source asset, that any returned path should be able to satisfy.
- **destination** (`Union[str, List[Asset]]`) – The destination account or the destination assets.

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

`ConnectionError`: if you have not successfully connected to the server.

`NotFoundError`: if `status_code == 404`

`BadRequestError`: if `400 <= status_code < 500` and `status_code != 404`

BadRequestError: if $500 \leq \text{status_code} < 600$

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters *cursor* (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit(*limit*)

Sets *limit* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters *limit* (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets *order* parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters *desc* (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

TradeAggregationsCallBuilder

```
class stellar_sdk.call_builder.call_builder_async.TradeAggregationsCallBuilder(horizon_url,  
                                                                           client, base,  
                                                                           counter,  
                                                                           resolution,  
                                                                           start_time=None,  
                                                                           end_time=None,  
                                                                           offset=None)
```

Creates a new *TradeAggregationsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.trade_aggregations()`.

Trade Aggregations facilitate efficient gathering of historical trade data.

See [List Trade Aggregations](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (`BaseAsyncClient`) – The client instance used to send request.
- **base** (`Asset`) – base asset
- **counter** (`Asset`) – counter asset
- **resolution** (`int`) – segment duration as millis since epoch. *Supported values are 1 minute (60000), 5 minutes (300000), 15 minutes (900000), 1 hour (3600000), 1 day (86400000) and 1 week (604800000).*
- **start_time** (`Optional[int]`) – lower time boundary represented as millis since epoch
- **end_time** (`Optional[int]`) – upper time boundary represented as millis since epoch
- **offset** (`Optional[int]`) – segments can be offset using this parameter. Expressed in milliseconds. *Can only be used if the resolution is greater than 1 hour. Value must be in whole hours, less than the provided resolution, and less than 24 hours.*

async call()

Triggers a HTTP request using this builder’s current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor (`cursor`)

Sets `cursor` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

limit (`limit`)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (`int`) – Number of records the server should return.

Returns

order (`desc=True`)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (`bool`) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

TradesCallBuilder

class `stellar_sdk.call_builder.call_builder_async.TradesCallBuilder`(*horizon_url, client*)

Creates a new *TradesCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.trades()`.

See [List All Trades](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (*BaseAsyncClient*) – The client instance used to send request.

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return Coroutine.

Raises

- ConnectionError*: if you have not successfully connected to the server.
- NotFoundError*: if `status_code == 404`
- BadRequestError*: if `400 <= status_code < 500` and `status_code != 404`
- BadResponseError*: if `500 <= status_code < 600`
- UnknownRequestError*: if an unknown error occurs, please submit an issue

cursor(*cursor*)

Sets *cursor* parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current CallBuilder instance

for_account(*account_id*)

Filter trades for a specific account

See [Retrieve an Account's Trades](#) for more information.

Parameters **account_id** (`str`) – account id

Returns current TradesCallBuilder instance

for_asset_pair(*base, counter*)

Filter trades for a specific asset pair (orderbook)

See [List All Trades](#) for more information.

Parameters

- **base** (*Asset*) – base asset
- **counter** (*Asset*) – counter asset

Returns current TradesCallBuilder instance

for_liquidity_pool(*liquidity_pool_id*)

Filter trades for a specific liquidity pool.

See [Liquidity Pools - Retrieve related Trades](#)

Parameters **liquidity_pool_id** (*str*) – The ID of the liquidity pool in hex string.

Returns current TradesCallBuilder instance

for_offer(*offer_id*)

Filter trades for a specific offer

See [List All Trades](#) for more information.

Parameters **offer_id** (*Union[int, str]*) – offer id

Returns current TradesCallBuilder instance

for_trade_type(*trade_type*)

Filter trades for a specific trade type

Horizon will reject requests which attempt to set *trade_type* to `liquidity_pools` when using the offer id filter.

Parameters **trade_type** (*str*) – trade type, the currently supported types are "orderbook", "liquidity_pools" and "all", defaults to "all".

Returns current TradesCallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (*int*) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (*bool*) – Sort direction, True to get desc sort direction, the default setting is True.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an `EventSource`.

Raise `StreamClientError` - Failed to fetch stream resource.

TransactionsCallBuilder

class `stellar_sdk.call_builder.call_builder_async.TransactionsCallBuilder`(*horizon_url, client*)
Creates a new *TransactionsCallBuilder* pointed to server defined by *horizon_url*. Do not create this object directly, use `stellar_sdk.ServerAsync.transactions()`.

See [List All Transactions](#) for more information.

Parameters

- **horizon_url** (`str`) – Horizon server URL.
- **client** (`BaseAsyncClient`) – The client instance used to send request.

async call()

Triggers a HTTP request using this builder's current configuration.

Return type `Dict[str, Any]`

Returns If it is called synchronous, the response will be returned. If it is called asynchronously, it will return `Coroutine`.

Raises

ConnectionError: if you have not successfully connected to the server.

NotFoundError: if `status_code == 404`

BadRequestError: if `400 <= status_code < 500` and `status_code != 404`

BadResponseError: if `500 <= status_code < 600`

UnknownRequestError: if an unknown error occurs, please submit an issue

cursor(cursor)

Sets `cursor` parameter for the current call. Returns the `CallBuilder` object on which this method has been called.

See [Pagination](#)

Parameters **cursor** (`Union[int, str]`) – A cursor is a value that points to a specific location in a collection of resources.

Returns current `CallBuilder` instance

for_account(account_id)

This endpoint represents all transactions that affected a given account.

See [Retrieve an Account's Transactions](#) for more information.

Parameters **account_id** (`str`) – account id

Returns current `TransactionsCallBuilder` instance

for_claimable_balance(claimable_balance_id)

This endpoint represents all transactions referencing a given claimable balance and can be used in streaming mode.

See [Claimable Balances - Retrieve related Transactions](#)

Parameters **claimable_balance_id** (`str`) – This claimable balance's id encoded in a hex string representation.

Returns current TransactionsCallBuilder instance

for_ledger(*sequence*)

This endpoint represents all transactions in a given ledger.

See [Retrieve a Ledger's Transactions](#) for more information.

Parameters **sequence** (`Union[str, int]`) – ledger sequence

Returns current TransactionsCallBuilder instance

for_liquidity_pool(*liquidity_pool_id*)

This endpoint represents all transactions referencing a given liquidity pool.

See [Liquidity Pools - Retrieve related Transactions](#)

Parameters **liquidity_pool_id** (`str`) – The ID of the liquidity pool in hex string.

Returns this TransactionsCallBuilder instance

include_failed(*include_failed*)

Adds a parameter defining whether to include failed transactions. By default only transactions of successful transactions are returned.

Parameters **include_failed** (`bool`) – Set to *True* to include failed transactions.

Returns current TransactionsCallBuilder instance

limit(*limit*)

Sets `limit` parameter for the current call. Returns the CallBuilder object on which this method has been called.

See [Pagination](#)

Parameters **limit** (`int`) – Number of records the server should return.

Returns

order(*desc=True*)

Sets `order` parameter for the current call. Returns the CallBuilder object on which this method has been called.

Parameters **desc** (`bool`) – Sort direction, *True* to get desc sort direction, the default setting is *True*.

Returns current CallBuilder instance

stream()

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Return type `AsyncGenerator[Dict[str, Any], None]`

Returns an EventSource.

Raise `StreamClientError` - Failed to fetch stream resource.

transaction(*transaction_hash*)

The transaction details endpoint provides information on a single transaction. The transaction hash provided in the hash argument specifies which transaction to load.

See [Retrieve a Transaction](#) for more information.

Parameters **transaction_hash** (`str`) – transaction hash

Returns current TransactionsCallBuilder instance

2.1.5 Client

BaseAsyncClient

class stellar_sdk.client.base_async_client.**BaseAsyncClient**

This is an abstract class, and if you want to implement your own asynchronous client, you **must** implement this class.

abstract async get(url, params=None)

Perform HTTP GET request.

Parameters

- **url** (*str*) – the request url
- **params** (*Optional[Dict[str, str]]*) – the request params

Return type *Response*

Returns the response from server

Raise *ConnectionError*

abstract async post(url, data)

Perform HTTP POST request.

Parameters

- **url** (*str*) – the request url
- **data** (*Dict[str, str]*) – the data send to server

Return type *Response*

Returns the response from server

Raise *ConnectionError*

abstract async stream(url, params=None)

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Parameters

- **url** (*str*) – the request url
- **params** (*Optional[Dict[str, str]]*) – the request params

Return type *AsyncGenerator[Dict[str, Any], None]*

Returns a dict AsyncGenerator for server response

Raise *ConnectionError*

BaseSyncClient

class `stellar_sdk.client.base_sync_client.BaseSyncClient`

This is an abstract class, and if you want to implement your own synchronous client, you **must** implement this class.

abstract `get(url, params=None)`

Perform HTTP GET request.

Parameters

- `url` (`str`) – the request url
- `params` (`Optional[Dict[str, str]]`) – the request params

Return type `Response`

Returns the response from server

Raise `ConnectionError`

abstract `post(url, data)`

Perform HTTP POST request.

Parameters

- `url` (`str`) – the request url
- `data` (`Dict[str, str]`) – the data send to server

Return type `Response`

Returns the response from server

Raise `ConnectionError`

abstract `stream(url, params=None)`

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Parameters

- `url` (`str`) – the request url
- `params` (`Optional[Dict[str, str]]`) – the request params

Return type `Generator[Dict[str, Any], None, None]`

Returns a dict Generator for server response

Raise `ConnectionError`

AiohttpClient

```
class stellar_sdk.client.aiohttp_client.AiohttpClient(pool_size=None, request_timeout=11,
                                                    post_timeout=33.0, backoff_factor=0.5,
                                                    user_agent=None, **kwargs)
```

The *AiohttpClient* object is a asynchronous http client, which represents the interface for making requests to a server instance.

Parameters

- **pool_size** (*Optional[int]*) – persistent connection to Horizon and connection pool
- **request_timeout** (*float*) – the timeout for all GET requests
- **post_timeout** (*float*) – the timeout for all POST requests
- **backoff_factor** (*Optional[float]*) – a backoff factor to apply between attempts after the second try
- **user_agent** (*Optional[str]*) – the server can use it to identify you

async close()

Close underlying connector.

Release all acquired resources.

Return type *None*

async get(url, params=None)

Perform HTTP GET request.

Parameters

- **url** (*str*) – the request url
- **params** (*Optional[Dict[str, str]]*) – the request params

Return type *Response*

Returns the response from server

Raise *ConnectionError*

async post(url, data=None)

Perform HTTP POST request.

Parameters

- **url** (*str*) – the request url
- **data** (*Optional[Dict[str, str]]*) – the data send to server

Return type *Response*

Returns the response from server

Raise *ConnectionError*

stream(url, params=None)

Perform Stream request.

Parameters

- **url** (*str*) – the request url
- **params** (*Optional[Dict[str, str]]*) – the request params

Return type *AsyncGenerator[Dict[str, Any], None]*

Returns the stream response from server

Raise `StreamClientError` - Failed to fetch stream resource.

RequestsClient

```
class stellar_sdk.client.requests_client.RequestsClient(pool_size=10, num_retries=3,
                                                       request_timeout=11, post_timeout=33.0,
                                                       backoff_factor=0.5, session=None,
                                                       stream_session=None)
```

The `RequestsClient` object is a synchronous http client, which represents the interface for making requests to a server instance.

Parameters

- **pool_size** (`int`) – persistent connection to Horizon and connection pool
- **num_retries** (`int`) – configurable request retry functionality
- **request_timeout** (`int`) – the timeout for all GET requests
- **post_timeout** (`float`) – the timeout for all POST requests
- **backoff_factor** (`float`) – a backoff factor to apply between attempts after the second try
- **session** (`Optional[Session]`) – the request session
- **stream_session** (`Optional[Session]`) – the stream request session

`close()`

Close underlying connector.

Release all acquired resources.

Return type `None`

`get(url, params=None)`

Perform HTTP GET request.

Parameters

- **url** (`str`) – the request url
- **params** (`Optional[Dict[str, str]]`) – the request params

Return type `Response`

Returns the response from server

Raise `ConnectionError`

`post(url, data=None)`

Perform HTTP POST request.

Parameters

- **url** (`str`) – the request url
- **data** (`Optional[Dict[str, str]]`) – the data send to server

Return type `Response`

Returns the response from server

Raise `ConnectionError`

stream(*url*, *params=None*)

Creates an EventSource that listens for incoming messages from the server.

See [Horizon Response Format](#)

See [MDN EventSource](#)

Parameters

- **url** (*str*) – the request url
- **params** (*Optional[Dict[str, str]]*) – the request params

Return type *Generator[Dict[str, Any], None, None]*

Returns a Generator for server response

Raise *ConnectionError*

SimpleRequestsClient

class `stellar_sdk.client.simple_requests_client.SimpleRequestsClient`

The *SimpleRequestsClient* object is a synchronous http client, which represents the interface for making requests to a server instance.

This client is to guide you in writing a client that suits your needs. I don't recommend that you actually use it.

get(*url*, *params=None*)

Perform HTTP GET request.

Parameters

- **url** (*str*) – the request url
- **params** (*Optional[Dict[str, str]]*) – the request params

Return type *Response*

Returns the response from server

Raise *ConnectionError*

post(*url*, *data*)

Perform HTTP POST request.

Parameters

- **url** (*str*) – the request url
- **data** (*Dict[str, str]*) – the data send to server

Return type *Response*

Returns the response from server

Raise *ConnectionError*

stream(*url*, *params=None*)

Not Implemented

Parameters

- **url** (*str*) – the request url
- **params** (*Optional[Dict[str, str]]*) – the request params

Return type `Generator[Dict[str, Any], None, None]`

Returns `None`

Response

class `stellar_sdk.client.response.Response`(*status_code, text, headers, url*)
The *Response* object, which contains a server's response to an HTTP request.

Parameters

- **status_code** (`int`) – response status code
- **text** (`str`) – response content
- **headers** (`dict`) – response headers
- **url** (`str`) – request url

`json()`

convert the content to dict

Return type `dict`

Returns the content from server

2.1.6 Exceptions

SdkError

class `stellar_sdk.exceptions.SdkError`
Base exception for all stellar sdk related errors

BadSignatureError

class `stellar_sdk.exceptions.BadSignatureError`
Raised when the signature was forged or otherwise corrupt.

Ed25519PublicKeyInvalidError

class `stellar_sdk.exceptions.Ed25519PublicKeyInvalidError`
Ed25519 public key is incorrect.

Ed25519SecretSeedInvalidError

class `stellar_sdk.exceptions.Ed25519SecretSeedInvalidError`
Ed25519 secret seed is incorrect.

MissingEd25519SecretSeedError

```
class stellar_sdk.exceptions.MissingEd25519SecretSeedError
    Missing Ed25519 secret seed in the keypair
```

MemoInvalidException

```
class stellar_sdk.exceptions.MemoInvalidException
    Memo is incorrect.
```

AssetCodeInvalidError

```
class stellar_sdk.exceptions.AssetCodeInvalidError
    Asset Code is incorrect.
```

AssetIssuerInvalidError

```
class stellar_sdk.exceptions.AssetIssuerInvalidError
    Asset issuer is incorrect.
```

NoApproximationError

```
class stellar_sdk.exceptions.NoApproximationError
    Approximation cannot be found
```

SignatureExistError

```
class stellar_sdk.exceptions.SignatureExistError
    A keypair can only sign a transaction once.
```

BaseRequestError

```
class stellar_sdk.exceptions.BaseRequestError
    Base class for requests errors.
```

ConnectionError

```
class stellar_sdk.exceptions.ConnectionError
    Base class for client connection errors.
```

BaseHorizonError

class stellar_sdk.exceptions.**BaseHorizonError**(*response*)
 Base class for horizon request errors.

Parameters **response** (*Response*) – client response

NotFoundError

class stellar_sdk.exceptions.**NotFound**(*response*)
 This exception is thrown when the requested resource does not exist. status_code == 400

BadRequestError

class stellar_sdk.exceptions.**BadRequestError**(*response*)
 The request from the client has an error. 400 <= status_code < 500 and status_code != 404

BadResponseError

class stellar_sdk.exceptions.**BadResponseError**(*response*)
 The response from the server has an error. 500 <= status_code < 600

FeatureNotEnabledError

class stellar_sdk.exceptions.**FeatureNotEnabledError**
 The feature is not enabled.

2.1.7 Keypair

class stellar_sdk.keypair.**Keypair**(*verify_key, signing_key=None*)
 The *Keypair* object, which represents a signing and verifying key for use with the Stellar network.

Instead of instantiating the class directly, we recommend using one of several class methods:

- *Keypair.random()*
- *Keypair.from_secret()*
- *Keypair.from_public_key()*

Learn how to create a key through our documentation: [Generate Keypair](#).

Parameters

- **verify_key** (*VerifyKey*) – The verifying (public) Ed25519 key in the keypair.
- **signing_key** (*Optional[SigningKey]*) – The signing (private) Ed25519 key in the keypair.

can_sign()

Returns *True* if this *Keypair* object contains secret key and can sign.

Return type *bool*

Returns *True* if this *Keypair* object contains secret key and can sign

classmethod `from_mnemonic_phrase`(*cls*, *mnemonic_phrase*, *language=Language.ENGLISH*, *passphrase=""*, *index=0*)

Generate a *Keypair* object via a mnemonic phrase.

Parameters

- **mnemonic_phrase** (*str*) – A unique string used to deterministically generate keypairs.
- **language** (*Union[Language, str]*) – The language of the mnemonic phrase, defaults to english.
- **passphrase** (*str*) – An optional passphrase used as part of the salt during PBKDF2 rounds when generating the seed from the mnemonic.
- **index** (*int*) – The index of the keypair generated by the mnemonic. This allows for multiple Keypairs to be derived from the same mnemonic, such as:

```
>>> from stellar_sdk.keypair import Keypair
>>> mnemonic = 'update hello cry airport drive chunk elite boat_
↳shaft sea describe number' # Don't use this mnemonic in practice.
>>> kp1 = Keypair.from_mnemonic_phrase(mnemonic, index=0)
>>> kp2 = Keypair.from_mnemonic_phrase(mnemonic, index=1)
>>> kp3 = Keypair.from_mnemonic_phrase(mnemonic, index=2)
```

Return type *Keypair*

Returns A new *Keypair* object derived from the mnemonic.

classmethod `from_public_key`(*cls*, *public_key*)

Generate a *Keypair* object from a public key.

Parameters **public_key** (*str*) – public key (ex. "GATPGGOIE6VWADVVD3ER3IF02IH6DTH6D5G535ITB3TT66FZFSIZE...")

Return type *Keypair*

Returns A new *Keypair* object derived by the public key.

Raise *Ed25519PublicKeyInvalidError*: if *public_key* is not a valid ed25519 public key.

classmethod `from_raw_ed25519_public_key`(*cls*, *raw_public_key*)

Generate a *Keypair* object from ed25519 public key raw bytes.

Parameters **raw_public_key** (*bytes*) – ed25519 public key raw bytes

Return type *Keypair*

Returns A new *Keypair* object derived by the ed25519 public key raw bytes

classmethod `from_raw_ed25519_seed`(*cls*, *raw_seed*)

Generate a *Keypair* object from ed25519 secret key seed raw bytes.

Parameters **raw_seed** (*bytes*) – ed25519 secret key seed raw bytes

Return type *Keypair*

Returns A new *Keypair* object derived by the ed25519 secret key seed raw bytes

classmethod `from_secret`(*cls*, *secret*)

Generate a *Keypair* object from a secret key.

Parameters **secret** (*str*) – secret key (ex. "SB2LHKBL24ITV2Y346BU46XPEL45BDAF00JLZ6SESCJZ6V5JMP7D6G5X...")

Return type *Keypair*

Returns A new *Keypair* object derived by the secret.

Raise *Ed25519SecretSeedInvalidError*: if *secret* is not a valid ed25519 secret seed.

static generate_mnemonic_phrase(*language=Language.ENGLISH, strength=128*)

Generate a mnemonic phrase.

Parameters

- **language** (*Union[Language, str]*) – The language of the mnemonic phrase, defaults to english.
- **strength** (*int*) – The complexity of the mnemonic, its possible value is 128, 160, 192, 224 and 256.

Return type *str*

Returns A mnemonic phrase.

property public_key: *str*

Returns public key associated with this *Keypair* object

Return type *str*

Returns public key

classmethod random(*cls*)

Generate a *Keypair* object from a randomly generated seed.

Return type *Keypair*

Returns A new *Keypair* object derived by the randomly seed.

raw_public_key()

Returns raw public key.

Return type *bytes*

Returns raw public key

raw_secret_key()

Returns raw secret key.

Return type *bytes*

Returns raw secret key

property secret: *str*

Returns secret key associated with this *Keypair* object

Return type *str*

Returns secret key

Raise *MissingEd25519SecretSeedError* The *Keypair* does not contain secret seed

sign(*data*)

Sign the provided data with the keypair's private key.

Parameters *data* (*bytes*) – The data to sign.

Return type *bytes*

Returns signed bytes

Raise *MissingEd25519SecretSeedError*: if *Keypair* does not contain secret seed.

sign_decorated(*data*)

Sign the provided data with the keypair's private key and returns DecoratedSignature.

Parameters `data` (*bytes*) – signed bytes

Return type *DecoratedSignature*

Returns sign decorated

signature_hint()

Returns signature hint associated with this *Keypair* object

Return type *bytes*

Returns signature hint

verify(*data*, *signature*)

Verify the provided data and signature match this keypair's public key.

Parameters

- **data** (*bytes*) – The data that was signed.
- **signature** (*bytes*) – The signature.

Raise *BadSignatureError*: if the verification failed and the signature was incorrect.

Return type *None*

xdr_public_key()

Return type *PublicKey*

Returns xdr public key

2.1.8 LiquidityPoolAsset

`stellar_sdk.liquidity_pool_asset.LIQUIDITY_POOL_FEE_V18 = 30`

`LIQUIDITY_POOL_FEE_V18` is the default liquidity pool fee in protocol v18. It defaults to 30 base points (0.3%).

class `stellar_sdk.liquidity_pool_asset.LiquidityPoolAsset(asset_a, asset_b, fee=30)`

The *LiquidityPoolAsset* object, which represents a liquidity pool trustline change.

Parameters

- **asset_a** (*Asset*) – The first asset in the Pool, it must respect the rule `asset_a < asset_b`. See `stellar_sdk.liquidity_pool_asset.LiquidityPoolAsset.is_valid_lexicographic_order()` for more details on how assets are sorted.
- **asset_b** (*Asset*) – The second asset in the Pool, it must respect the rule `asset_a < asset_b`. See `stellar_sdk.liquidity_pool_asset.LiquidityPoolAsset.is_valid_lexicographic_order()` for more details on how assets are sorted.
- **fee** (*int*) – The liquidity pool fee. For now the only fee supported is `30`.

Raise *ValueError*

classmethod `from_xdr_object(cls, xdr_object)`

Create a *LiquidityPoolAsset* from an XDR *ChangeTrustAsset* object.

Parameters `xdr_object` (*ChangeTrustAsset*) – The XDR *ChangeTrustAsset* object.

Return type *LiquidityPoolAsset*

Returns A new *LiquidityPoolAsset* object from the given XDR *ChangeTrustAsset* object.

static `is_valid_lexicographic_order(asset_a, asset_b)`

Compares if `asset_a < asset_b` according with the criteria:

1. First compare the type (eg. native before alphanum4 before alphanum12).
2. If the types are equal, compare the assets codes.
3. If the asset codes are equal, compare the issuers.

Parameters

- **asset_a** (*Asset*) – The first asset in the lexicographic order.
- **asset_b** (*Asset*) – The second asset in the lexicographic order.

Return type `bool`

Returns return `True` if `asset_a < asset_b`

property `liquidity_pool_id: str`

Computes the liquidity pool id for current instance.

Return type `str`

Returns Liquidity pool id.

to_change_trust_asset_xdr_object()

Returns the xdr object for this `ChangeTrustAsset` object.

Return type *ChangeTrustAsset*

Returns XDR `ChangeTrustAsset` object

2.1.9 LiquidityPoolId

class `stellar_sdk.liquidity_pool_id.LiquidityPoolId(liquidity_pool_id)`

The *LiquidityPoolId* object, which represents the asset referenced by a trustline to a liquidity pool.

Parameters `liquidity_pool_id (str)` – The ID of the liquidity pool in hex string.

Raise `ValueError`

classmethod `from_xdr_object(cls, xdr_object)`

Create a *LiquidityPoolId* from an XDR `Asset` object.

Parameters `xdr_object (TrustLineAsset)` – The XDR `TrustLineAsset` object.

Return type *LiquidityPoolId*

Returns A new *LiquidityPoolId* object from the given XDR `TrustLineAsset` object.

to_trust_line_asset_xdr_object()

Returns the xdr object for this `LiquidityPoolId` object.

Return type *TrustLineAsset*

Returns XDR `TrustLineAsset` object

2.1.10 Memo

Memo

class stellar_sdk.memo.Memo

The *Memo* object, which represents the base class for memos for use with Stellar transactions.

The memo for a transaction contains optional extra information about the transaction taking place. It is the responsibility of the client to interpret this value.

See the following implementations that serve a more practical use with the library:

- *NoneMemo* - No memo.
- *TextMemo* - A string encoded using either ASCII or UTF-8, up to 28-bytes long.
- *IdMemo* - A 64 bit unsigned integer.
- *HashMemo* - A 32 byte hash.
- *RetHashMemo* - A 32 byte hash intended to be interpreted as the hash of the transaction the sender is refunding.

See [Stellar's documentation on Transactions](#) for more information on how memos are used within transactions, as well as information on the available types of memos.

static from_xdr_object(*xdr_object*)

Returns an Memo object from XDR memo object.

Return type *Memo*

abstract to_xdr_object()

Creates an XDR Memo object that represents this *Memo*.

Return type *Memo*

NoneMemo

class stellar_sdk.memo.NoneMemo

The *NoneMemo*, which represents no memo for a transaction.

classmethod from_xdr_object(*cls*, *xdr_object*)

Returns an *NoneMemo* object from XDR memo object.

Return type *NoneMemo*

to_xdr_object()

Creates an XDR Memo object that represents this *NoneMemo*.

Return type *Memo*

TextMemo

class stellar_sdk.memo.**TextMemo**(*text*)

The *TextMemo*, which represents MEMO_TEXT in a transaction.

Parameters *text* (`Union[str, bytes]`) – A string encoded using either ASCII or UTF-8, up to 28-bytes long. Note, *text* can be anything, see [this issue](#) for more information.

Raises *MemoInvalidException*: if *text* is not a valid text memo.

classmethod **from_xdr_object**(*cls, xdr_object*)

Returns an *TextMemo* object from XDR memo object.

Return type *TextMemo*

to_xdr_object()

Creates an XDR Memo object that represents this *TextMemo*.

Return type *Memo*

IdMemo

class stellar_sdk.memo.**IdMemo**(*memo_id*)

The *IdMemo* which represents MEMO_ID in a transaction.

Parameters *memo_id* (`int`) – A 64 bit unsigned integer.

Raises *MemoInvalidException*: if *id* is not a valid id memo.

classmethod **from_xdr_object**(*cls, xdr_object*)

Returns an *IdMemo* object from XDR memo object.

Return type *IdMemo*

to_xdr_object()

Creates an XDR Memo object that represents this *IdMemo*.

Return type *Memo*

HashMemo

class stellar_sdk.memo.**HashMemo**(*memo_hash*)

The *HashMemo* which represents MEMO_HASH in a transaction.

Parameters *memo_hash* (`Union[bytes, str]`) – A 32 byte hash hex encoded string.

Raises *MemoInvalidException*: if *memo_hash* is not a valid hash memo.

classmethod **from_xdr_object**(*cls, xdr_object*)

Returns an *HashMemo* object from XDR memo object.

Return type *HashMemo*

to_xdr_object()

Creates an XDR Memo object that represents this *HashMemo*.

Return type *Memo*

ReturnHashMemo

class stellar_sdk.memo.ReturnHashMemo(*memo_return*)

The *ReturnHashMemo* which represents MEMO_RETURN in a transaction.

MEMO_RETURN is typically used with refunds/returns over the network - it is a 32 byte hash intended to be interpreted as the hash of the transaction the sender is refunding.

Parameters *memo_return* (Union[bytes, str]) – A 32 byte hash or hex encoded string intended to be interpreted as the hash of the transaction the sender is refunding.

Raises *MemoInvalidException*: if *memo_return* is not a valid return hash memo.

classmethod from_xdr_object(*cls, xdr_object*)

Returns an *ReturnHashMemo* object from XDR memo object.

Return type *ReturnHashMemo*

to_xdr_object()

Creates an XDR Memo object that represents this *ReturnHashMemo*.

Return type *Memo*

2.1.11 MuxedAccount

class stellar_sdk.muxed_account.MuxedAccount(*account_id, account_muxed_id=None*)

The *MuxedAccount* object, which represents a multiplexed account on Stellar's network.

An example:

```
from stellar_sdk import MuxedAccount

account_id = "GAQAA5L65LSYH7CQ3VTJ7F3HHLGCL3DSLAR2Y47263D56MNNGHSQSTVY"
account_muxed_id = 1234
account_muxed =
↳ "MAQAA5L65LSYH7CQ3VTJ7F3HHLGCL3DSLAR2Y47263D56MNNGHSQSAAAAAAAAAAAE2LP26"

# generate account_muxed
muxed = MuxedAccount(account=account_id, account_muxed_id=1234) # account_muxed_id
↳ is optional.
print(f"account_muxed: {muxed.account_muxed}") # `account_muxed` returns ``None`` if
↳ `account_muxed_id` is ``None``.

# parse account_muxed
muxed = MuxedAccount.from_account(account_muxed)
print(f"account_id: {muxed.account_id}\n"
      f"account_muxed_id: {muxed.account_muxed_id}")
```

See [SEP-0023](#) for more information.

Parameters

- **account_id** (str) – ed25519 account id, for example: "GDGQVOKHW4VEJRU2TETD6DBRKEO5ERCNF353LW5WBFW3JJWQ2BRQ6KDD". It should be a string starting with G. If you want to build a MuxedAccount object using an address starting with M, please use `stellar_sdk.MuxedAccount.from_account()`.
- **account_muxed_id** (Optional[int]) – account multiplexing id (ex. 1234)

property account_muxed: Optional[str]

Get the multiplex address starting with M, return None if *account_id_id* is None.

Return type Optional[str]

classmethod from_account(cls, account)

Create a *MuxedAccount* object from account id or muxed account id.

Parameters *account* (str) – account id or muxed account id (ex. "GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD" or "MAAAAAAAAAAJURAAB2X52XFQP6FBXLGT6LWOOWMEXWHEWBDVRZ7V5WH34Y22MPFBHUHY")

Return type *MuxedAccount*

classmethod from_xdr_object(cls, xdr_object)

Create a *MuxedAccount* object from an XDR Asset object.

Parameters *xdr_object* (*MuxedAccount*) – The *MuxedAccount* object.

Return type *MuxedAccount*

Returns A new *MuxedAccount* object from the given XDR *MuxedAccount* object.

to_xdr_object()

Returns the xdr object for this *MuxedAccount* object.

Return type *MuxedAccount*

Returns XDR *MuxedAccount* object

property universal_account_id: str

Get the universal account id, if *account_muxed_id* is None, it will return ed25519 public key (ex. "GDGQVOKHW4VEJRU2TETD6DBRKE05ERCNF353LW5WBFW3JJWQ2BRQ6KDD"), otherwise it will return muxed account (ex. "MAAAAAAAAAAJURAAB2X52XFQP6FBXLGT6LWOOWMEXWHEWBDVRZ7V5WH34Y22MPFBHUHY")

Return type str

2.1.12 Network

class stellar_sdk.network.**Network**(network_passphrase)

The *Network* object, which represents a Stellar network.

This class represents such a stellar network such as the Public network and the Test network.

Parameters *network_passphrase* (str) – The passphrase for the network. (ex. "Public Global Stellar Network ; September 2015")

PUBLIC_NETWORK_PASSPHRASE: str = 'Public Global Stellar Network ; September 2015'

The Public network passphrase.

TESTNET_NETWORK_PASSPHRASE: str = 'Test SDF Network ; September 2015'

The Test network passphrase.

network_id()

Get the network ID of the network, which is an hash of the passphrase.

Return type bytes

Returns The network ID of the network.

classmethod public_network(cls)

Get the *Network* object representing the PUBLIC Network.

Return type *Network*

Returns PUBLIC Network

classmethod `testnet_network(cls)`

Get the *Network* object representing the TESTNET Network.

Return type *Network*

Returns TESTNET Network

2.1.13 Operation

Operation

class `stellar_sdk.operation.Operation(source=None)`

The *Operation* object, which represents an operation on Stellar's network.

An operation is an individual command that mutates Stellar's ledger. It is typically rolled up into a transaction (a transaction is a list of operations with additional metadata).

Operations are executed on behalf of the source account specified in the transaction, unless there is an override defined for the operation.

For more on operations, see [Stellar's documentation on operations](#) as well as [Stellar's List of Operations](#), which includes information such as the security necessary for a given operation, as well as information about when validity checks occur on the network.

The *Operation* class is typically not used, but rather one of its subclasses is typically included in transactions.

Parameters `source` (`Union[MuxedAccount, str, None]`) – The source account for the operation.

Defaults to the transaction's source account.

static `from_xdr_amount(value)`

Converts an str amount from an XDR amount object

Parameters `value` (`int`) – The amount to convert to a string from an XDR int64 amount.

Return type `str`

classmethod `from_xdr_object(cls, xdr_object)`

Create the appropriate *Operation* subclass from the XDR object.

Parameters `xdr_object` (`Operation`) – The XDR object to create an *Operation* (or subclass) instance from.

Return type *Operation*

static `get_source_from_xdr_obj(xdr_object)`

Get the source account from account the operation xdr object.

Parameters `xdr_object` (`Operation`) – the operation xdr object.

Return type `Optional[MuxedAccount]`

Returns The source account from account the operation xdr object.

static `to_xdr_amount(value)`

Converts an amount to the appropriate value to send over the network as a part of an XDR object.

Each asset amount is encoded as a signed 64-bit integer in the XDR structures. An asset amount unit (that which is seen by end users) is scaled down by a factor of ten million (10,000,000) to arrive at the native 64-bit integer representation. For example, the integer amount value 25,123,456 equals 2.5123456 units of the asset. This scaling allows for seven decimal places of precision in human-friendly amount units.

This static method correctly multiplies the value by the scaling factor in order to come to the integer value used in XDR structures.

See [Stellar's documentation on Asset Precision](#) for more information.

Parameters `value` (`Union[str, Decimal]`) – The amount to convert to an integer for XDR serialization.

Return type `int`

`to_xdr_object()`

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

AccountMerge

class `stellar_sdk.operation.AccountMerge`(*destination, source=None*)

The *AccountMerge* object, which represents a AccountMerge operation on Stellar's network.

Transfers the native balance (the amount of XLM an account holds) to another account and removes the source account from the ledger.

Threshold: High

See [Account Merge](#) for more information.

Parameters

- **destination** (`Union[MixedAccount, str]`) – Destination to merge the source account into.
- **source** (`Union[MixedAccount, str, None]`) – The source account for the operation. Defaults to the transaction's source account.

classmethod `from_xdr_object`(*cls, xdr_object*)

Creates a *AccountMerge* object from an XDR Operation object.

Return type *AccountMerge*

`to_xdr_object()`

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

AllowTrust

class `stellar_sdk.operation.AllowTrust`(*trustor, asset_code, authorize, source=None*)

The *AllowTrust* object, which represents a AllowTrust operation on Stellar's network.

Updates the authorized flag of an existing trustline. This can only be called by the issuer of a trustline's asset.

The issuer can only clear the authorized flag if the issuer has the AUTH_REVOCABLE_FLAG set. Otherwise, the issuer can only set the authorized flag.

Threshold: Low

See [Allow Trust](#) for more information.

Parameters

- **trustor** (`str`) – The trusting account (the one being authorized).
- **asset_code** (`str`) – The asset code being authorized.

- **authorize** (`Union[TrustLineEntryFlag, bool]`) – *True* to authorize the line, *False* to deauthorize. If you need further control, you can also use `stellar_sdk.operation.allow_trust.TrustLineEntryFlag`.
- **source** (`Union[MixedAccount, str, None]`) – The source account for the operation. Defaults to the transaction's source account.

classmethod `from_xdr_object(cls, xdr_object)`

Creates a `AllowTrust` object from an XDR Operation object.

Return type `AllowTrust`

to_xdr_object()

Creates an XDR Operation object that represents this `Operation`.

Return type `Operation`

class `stellar_sdk.operation.allow_trust.TrustLineEntryFlag(value)`

Indicates which flags to set. For details about the flags, please refer to the [CAP-0018](#).

- **UNAUTHORIZED_FLAG**: The account can hold a balance but cannot receive payments, send payments, maintain offers or manage offers
- **AUTHORIZED_FLAG**: The account can hold a balance, receive payments, send payments, maintain offers or manage offers
- **AUTHORIZED_TO_MAINTAIN_LIABILITIES_FLAG**: The account can hold a balance and maintain offers but cannot receive payments, send payments or manage offers

BumpSequence

class `stellar_sdk.operation.BumpSequence(bump_to, source=None)`

The `BumpSequence` object, which represents a BumpSequence operation on Stellar's network.

Bump sequence allows to bump forward the sequence number of the source account of the operation, allowing to invalidate any transactions with a smaller sequence number. If the specified bumpTo sequence number is greater than the source account's sequence number, the account's sequence number is updated with that value, otherwise it's not modified.

Threshold: Low

See [Bump Sequence](#) for more information.

Parameters

- **bump_to** (`int`) – Sequence number to bump to.
- **source** (`Union[MixedAccount, str, None]`) – The optional source account.

classmethod `from_xdr_object(cls, xdr_object)`

Creates a `BumpSequence` object from an XDR Operation object.

Return type `BumpSequence`

to_xdr_object()

Creates an XDR Operation object that represents this `Operation`.

Return type `Operation`

ChangeTrust

class stellar_sdk.operation.**ChangeTrust**(*asset*, *limit=None*, *source=None*)

The *ChangeTrust* object, which represents a ChangeTrust operation on Stellar's network.

Creates, updates, or deletes a trustline. For more on trustlines, please refer to the [assets documentation](#).

Threshold: Medium

See [Change Trust](#) for more information.

Parameters

- **asset** (`Union[Asset, LiquidityPoolAsset]`) – The asset for the trust line.
- **limit** (`Union[str, Decimal, None]`) – The limit for the asset, defaults to `max(int64(922337203685.4775807))`. If the limit is set to "0" it deletes the trustline.
- **source** (`Union[MixedAccount, str, None]`) – The source account for the operation. Defaults to the transaction's source account.

classmethod **from_xdr_object**(*cls*, *xdr_object*)

Creates a *ChangeTrust* object from an XDR Operation object.

Return type *ChangeTrust*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

CreateAccount

class stellar_sdk.operation.**CreateAccount**(*destination*, *starting_balance*, *source=None*)

The *CreateAccount* object, which represents a Create Account operation on Stellar's network.

This operation creates and funds a new account with the specified starting balance.

Threshold: Medium

See [Create Account](#) for more information.

Parameters

- **destination** (`str`) – Destination account ID to create an account for.
- **starting_balance** (`Union[str, Decimal]`) – Amount in XLM the account should be funded for. Must be greater than the [reserve balance](#) amount.
- **source** (`Union[MixedAccount, str, None]`) – The source account for the operation. Defaults to the transaction's source account.

classmethod **from_xdr_object**(*cls*, *xdr_object*)

Creates a *CreateAccount* object from an XDR Operation object.

Return type *CreateAccount*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

CreatePassiveSellOffer

class stellar_sdk.operation.**CreatePassiveSellOffer**(*selling, buying, amount, price, source=None*)

The *CreatePassiveSellOffer* object, which represents a CreatePassiveSellOffer operation on Stellar's network.

A passive sell offer is an offer that does not act on and take a reverse offer of equal price. Instead, they only take offers of lesser price. For example, if an offer exists to buy 5 BTC for 30 XLM, and you make a passive sell offer to buy 30 XLM for 5 BTC, your passive sell offer does not take the first offer.

Note that regular offers made later than your passive sell offer can act on and take your passive sell offer, even if the regular offer is of the same price as your passive sell offer.

Passive sell offers allow market makers to have zero spread. If you want to trade EUR for USD at 1:1 price and USD for EUR also at 1:1, you can create two passive sell offers so the two offers don't immediately act on each other.

Once the passive sell offer is created, you can manage it like any other offer using the manage offer operation - see *ManageBuyOffer* for more details.

Threshold: Medium

See [Create Passive Sell Offer](#) for more information.

Parameters

- **selling** (*Asset*) – What you're selling.
- **buying** (*Asset*) – What you're buying.
- **amount** (*Union[str, Decimal]*) – The total amount you're selling.
- **price** (*Union[Price, str, Decimal]*) – Price of 1 unit of *selling* in terms of *buying*.
- **source** (*Union[MuxedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

classmethod **from_xdr_object**(*cls, xdr_object*)

Creates a *CreatePassiveSellOffer* object from an XDR Operation object.

Return type *CreatePassiveSellOffer*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

Inflation

class stellar_sdk.operation.**Inflation**(*source=None*)

The *Inflation* object, which represents a Inflation operation on Stellar's network.

This operation runs inflation.

Threshold: Low

Parameters **source** (*Union[MuxedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

classmethod **from_xdr_object**(*cls, xdr_object*)

Creates a *Inflation* object from an XDR Operation object.

Return type *Inflation*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

LiquidityPoolDeposit

class stellar_sdk.operation.LiquidityPoolDeposit(*liquidity_pool_id, max_amount_a, max_amount_b, min_price, max_price, source=None*)

The *LiquidityPoolDeposit* object, which represents a LiquidityPoolDeposit operation on Stellar's network.

Creates a liquidity pool deposit operation.

Threshold: Medium

See [Liquidity Pool Deposit](#) for more information.

Parameters

- **liquidity_pool_id** (*str*) – The liquidity pool ID.
- **max_amount_a** (*Union[str, Decimal]*) – Maximum amount of first asset to deposit.
- **max_amount_b** (*Union[str, Decimal]*) – Maximum amount of second asset to deposit.
- **min_price** (*Union[str, Decimal, Price]*) – Minimum deposit_a/deposit_b price.
- **max_price** (*Union[str, Decimal, Price]*) – Maximum deposit_a/deposit_b price.
- **source** (*Union[MuxedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

classmethod from_xdr_object(*cls, xdr_object*)

Creates a *LiquidityPoolDeposit* object from an XDR Operation object.

Return type *LiquidityPoolDeposit*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

LiquidityPoolWithdraw

class stellar_sdk.operation.LiquidityPoolWithdraw(*liquidity_pool_id, amount, min_amount_a, min_amount_b, source=None*)

The *LiquidityPoolWithdraw* object, which represents a LiquidityPoolWithdraw operation on Stellar's network.

Creates a liquidity pool withdraw operation.

Threshold: Medium

See [Liquidity Pool Withdraw](#) for more information.

Parameters

- **liquidity_pool_id** (*str*) – The liquidity pool ID.
- **amount** (*Union[str, Decimal]*) – Amount of pool shares to withdraw.
- **min_amount_a** (*Union[str, Decimal]*) – Minimum amount of first asset to withdraw.
- **min_amount_b** (*Union[str, Decimal]*) – Minimum amount of second asset to withdraw.

- **source** (`Union[MixedAccount, str, None]`) – The source account for the operation. Defaults to the transaction’s source account.

classmethod `from_xdr_object(cls, xdr_object)`

Creates a `LiquidityPoolWithdraw` object from an XDR Operation object.

Return type `LiquidityPoolWithdraw`

to_xdr_object()

Creates an XDR Operation object that represents this `Operation`.

Return type `Operation`

ManageBuyOffer

class `stellar_sdk.operation.ManageBuyOffer(selling, buying, amount, price, offer_id=0, source=None)`

The `ManageBuyOffer` object, which represents a ManageBuyOffer operation on Stellar’s network.

Creates, updates, or deletes an buy offer.

If you want to create a new offer set `offer_id` to 0.

If you want to update an existing offer set `offer_id` to existing offer ID.

If you want to delete an existing offer set `offer_id` to existing offer ID and set `amount` to 0.

Threshold: Medium

See [Manage Buy Offer](#) for more information.

Parameters

- **selling** (`Asset`) – What you’re selling.
- **buying** (`Asset`) – What you’re buying.
- **amount** (`Union[str, Decimal]`) – Amount being bought. if set to 0, delete the offer.
- **price** (`Union[Price, str, Decimal]`) – Price of thing being bought in terms of what you are selling.
- **offer_id** (`int`) – If 0, will create a new offer (default). Otherwise, edits an existing offer.
- **source** (`Union[MixedAccount, str, None]`) – The source account for the operation. Defaults to the transaction’s source account.

classmethod `from_xdr_object(cls, xdr_object)`

Creates a `ManageBuyOffer` object from an XDR Operation object.

Return type `ManageBuyOffer`

to_xdr_object()

Creates an XDR Operation object that represents this `Operation`.

Return type `Operation`

ManageData

class stellar_sdk.operation.**ManageData**(*data_name*, *data_value*, *source=None*)

The *ManageData* object, which represents a ManageData operation on Stellar’s network.

Allows you to set, modify or delete a Data Entry (name/value pair) that is attached to a particular account. An account can have an arbitrary amount of DataEntries attached to it. Each DataEntry increases the minimum balance needed to be held by the account.

DataEntries can be used for application specific things. They are not used by the core Stellar protocol.

Threshold: Medium

See [Manage Data](#) for more information.

Parameters

- **data_name** (*str*) – If this is a new Name it will add the given name/value pair to the account. If this Name is already present then the associated value will be modified. Up to 64 bytes long.
- **data_value** (*Union[str, bytes, None]*) – If not present then the existing *data_name* will be deleted. If present then this value will be set in the DataEntry. Up to 64 bytes long.
- **source** (*Union[MixedAccount, str, None]*) – The optional source account.

classmethod **from_xdr_object**(*cls*, *xdr_object*)

Creates a *ManageData* object from an XDR Operation object.

Return type *ManageData*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

ManageSellOffer

class stellar_sdk.operation.**ManageSellOffer**(*selling*, *buying*, *amount*, *price*, *offer_id=0*, *source=None*)

The *ManageSellOffer* object, which represents a ManageSellOffer operation on Stellar’s network.

Creates, updates, or deletes an sell offer.

If you want to create a new offer set *offer_id* to 0.

If you want to update an existing offer set *offer_id* to existing offer ID.

If you want to delete an existing offer set *offer_id* to existing offer ID and set *amount* to 0.

Threshold: Medium

See [Manage Sell Offer](#) for more information.

Parameters

- **selling** (*Asset*) – What you’re selling.
- **buying** (*Asset*) – What you’re buying.
- **amount** (*Union[str, Decimal]*) – The total amount you’re selling. If 0, deletes the offer.
- **price** (*Union[Price, str, Decimal]*) – Price of 1 unit of *selling* in terms of *buying*.
- **offer_id** (*int*) – If 0, will create a new offer (default). Otherwise, edits an existing offer.

- **source** (`Union[MixedAccount, str, None]`) – The source account for the operation. Defaults to the transaction’s source account.

classmethod `from_xdr_object(cls, xdr_object)`

Creates a `ManageSellOffer` object from an XDR Operation object.

Return type `ManageSellOffer`

to_xdr_object()

Creates an XDR Operation object that represents this `Operation`.

Return type `Operation`

PathPaymentStrictReceive

class `stellar_sdk.operation.PathPaymentStrictReceive(destination, send_asset, send_max, dest_asset, dest_amount, path, source=None)`

The `PathPaymentStrictReceive` object, which represents a PathPaymentStrictReceive operation on Stellar’s network.

Sends an amount in a specific asset to a destination account through a path of offers. This allows the asset sent (e.g. 450 XLM) to be different from the asset received (e.g. 6 BTC).

Threshold: Medium

See [Path Payment Strict Receive](#) for more information.

Parameters

- **destination** (`Union[MixedAccount, str]`) – The destination account to send to.
- **send_asset** (`Asset`) – The *asset* to pay with.
- **send_max** (`Union[str, Decimal]`) – The maximum amount of *send_asset* to send.
- **dest_asset** (`Asset`) – The asset the *destination* will receive.
- **dest_amount** (`Union[str, Decimal]`) – The amount the *destination* receives.
- **path** (`List[Asset]`) – A list of `Asset` objects to use as the path.
- **source** (`Union[MixedAccount, str, None]`) – The source account for the operation. Defaults to the transaction’s source account.

classmethod `from_xdr_object(cls, xdr_object)`

Creates a `PathPaymentStrictReceive` object from an XDR Operation object.

Return type `PathPaymentStrictReceive`

to_xdr_object()

Creates an XDR Operation object that represents this `Operation`.

Return type `Operation`

PathPaymentStrictSend

class stellar_sdk.operation.**PathPaymentStrictSend**(*destination, send_asset, send_amount, dest_asset, dest_min, path, source=None*)

The *PathPaymentStrictSend* object, which represents a PathPaymentStrictSend operation on Stellar's network.

Sends an amount in a specific asset to a destination account through a path of offers. This allows the asset sent (e.g, 450 XLM) to be different from the asset received (e.g, 6 BTC).

Threshold: Medium

See [Path Payment Strict Send](#) for more information.

Parameters

- **destination** ([Union\[MixedAccount, str\]](#)) – The destination account to send to.
- **send_asset** ([Asset](#)) – The *asset* to pay with.
- **send_amount** ([Union\[str, Decimal\]](#)) – Amount of *send_asset* to send.
- **dest_asset** ([Asset](#)) – The asset the *destination* will receive.
- **dest_min** ([Union\[str, Decimal\]](#)) – The minimum amount of *dest_asset* to be received.
- **path** ([List\[Asset\]](#)) – A list of [Asset](#) objects to use as the path.
- **source** ([Union\[MixedAccount, str, None\]](#)) – The source account for the operation. Defaults to the transaction's source account.

classmethod **from_xdr_object**(*cls, xdr_object*)

Creates a *PathPaymentStrictSend* object from an XDR Operation object.

Return type *PathPaymentStrictSend*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

Payment

class stellar_sdk.operation.**Payment**(*destination, asset, amount, source=None*)

The *Payment* object, which represents a Payment operation on Stellar's network.

Sends an amount in a specific asset to a destination account.

Threshold: Medium

See [Payment](#) for more information.

Parameters

- **destination** ([Union\[MixedAccount, str\]](#)) – The destination account ID.
- **asset** ([Asset](#)) – The asset to send.
- **amount** ([Union\[str, Decimal\]](#)) – The amount to send.
- **source** ([Union\[MixedAccount, str, None\]](#)) – The source account for the operation. Defaults to the transaction's source account.

classmethod **from_xdr_object**(*cls, xdr_object*)

Creates a *Payment* object from an XDR Operation object.

Return type *Payment*

`to_xdr_object()`

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

SetOptions

```
class stellar_sdk.operation.SetOptions(inflation_dest=None, clear_flags=None, set_flags=None,  
master_weight=None, low_threshold=None,  
med_threshold=None, high_threshold=None, signer=None,  
home_domain=None, source=None)
```

The *SetOptions* object, which represents a SetOptions operation on Stellar's network.

This operation sets the options for an account.

For more information on the signing options, please refer to the [multi-sig doc](#).

When updating signers or other thresholds, the threshold of this operation is high.

Threshold: Medium or High

See [Set Options](#) for more information.

Parameters

- **inflation_dest** (`Optional[str]`) – Account of the inflation destination.
- **clear_flags** (`Union[int, AuthorizationFlag, None]`) – Indicates which flags to clear. For details about the flags, please refer to the [Control Access to an Asset - Flag](#). The bit mask integer subtracts from the existing flags of the account. This allows for setting specific bits without knowledge of existing flags, you can also use `stellar_sdk.operation.set_options.AuthorizationFlag`
 - AUTHORIZATION_REQUIRED = 1
 - AUTHORIZATION_REVOCABLE = 2
 - AUTHORIZATION_IMMUTABLE = 4
 - AUTHORIZATION_CLAWBACK_ENABLED = 8
- **set_flags** (`Union[int, AuthorizationFlag, None]`) – Indicates which flags to set. For details about the flags, please refer to the [Control Access to an Asset - Flag](#). The bit mask integer adds onto the existing flags of the account. This allows for setting specific bits without knowledge of existing flags, you can also use `stellar_sdk.operation.set_options.AuthorizationFlag`
 - AUTHORIZATION_REQUIRED = 1
 - AUTHORIZATION_REVOCABLE = 2
 - AUTHORIZATION_IMMUTABLE = 4
 - AUTHORIZATION_CLAWBACK_ENABLED = 8
- **master_weight** (`Optional[int]`) – A number from 0-255 (inclusive) representing the weight of the master key. If the weight of the master key is updated to 0, it is effectively disabled.
- **low_threshold** (`Optional[int]`) – A number from 0-255 (inclusive) representing the threshold this account sets on all operations it performs that have a [low threshold](#).

- **med_threshold** (*Optional[int]*) – A number from 0-255 (inclusive) representing the threshold this account sets on all operations it performs that have a [medium threshold](#).
- **high_threshold** (*Optional[int]*) – A number from 0-255 (inclusive) representing the threshold this account sets on all operations it performs that have a [high threshold](#).
- **home_domain** (*Optional[str]*) – sets the home domain used for reverse [federation](#) lookup.
- **signer** (*Optional[Signer]*) – Add, update, or remove a signer from the account.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction’s source account.

classmethod `from_xdr_object(cls, xdr_object)`

Creates a *SetOptions* object from an XDR Operation object.

Return type *SetOptions*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

class `stellar_sdk.operation.set_options.AuthorizationFlag(value)`

Indicates which flags to set. For details about the flags, please refer to the [Control Access to an Asset - Flag](#).

CreateClaimableBalance

class `stellar_sdk.operation.CreateClaimableBalance(asset, amount, claimants, source=None)`

The *CreateClaimableBalance* object, which represents a CreateClaimableBalance operation on Stellar’s network.

Creates a ClaimableBalanceEntry. See [Claimable Balance](#) for more information on parameters and usage.

Threshold: Medium

See [Create Claimable Balance](#) for more information.

Parameters

- **asset** (*Asset*) – The asset for the claimable balance.
- **amount** (*Union[str, Decimal]*) – the amount of the asset.
- **claimants** (*List[Claimant]*) – A list of Claimants.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction’s source account.

classmethod `from_xdr_object(cls, xdr_object)`

Creates a *CreateClaimableBalance* object from an XDR Operation object.

Return type *CreateClaimableBalance*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

class `stellar_sdk.operation.Claimant(destination, predicate=None)`

The *Claimant* object represents a claimable balance claimant.

Parameters

- **destination** (*str*) – The destination account ID.

- **predicate** (*Optional[ClaimPredicate]*) – The claim predicate. It is optional, it defaults to unconditional if none is specified.

class stellar_sdk.operation.ClaimPredicate(*claim_predicate_type, and_predicates, or_predicates, not_predicate, abs_before, rel_before*)

The *ClaimPredicate* object, which represents a ClaimPredicate on Stellar’s network.

We do not recommend that you build it through the constructor, please use the helper function.

Parameters

- **claim_predicate_type** (*ClaimPredicateType*) – Type of ClaimPredicate.
- **and_predicates** (*Optional[ClaimPredicateGroup]*) – The ClaimPredicates.
- **or_predicates** (*Optional[ClaimPredicateGroup]*) – The ClaimPredicates.
- **not_predicate** (*Optional[ClaimPredicate]*) – The ClaimPredicate.
- **abs_before** (*Optional[int]*) – Unix epoch.
- **rel_before** (*Optional[int]*) – seconds since closeTime of the ledger in which the ClaimableBalanceEntry was created.

classmethod predicate_and(*cls, left, right*)

Returns an **and** claim predicate

Parameters

- **left** (*ClaimPredicate*) – a ClaimPredicate.
- **right** (*ClaimPredicate*) – a ClaimPredicate.

Return type *ClaimPredicate*

Returns an **and** claim predicate.

classmethod predicate_before_absolute_time(*cls, abs_before*)

Returns a **before_absolute_time** claim predicate.

This predicate will be fulfilled if the closing time of the ledger that includes the *CreateClaimableBalance* operation is less than this (absolute) Unix timestamp.

Parameters **abs_before** (*int*) – Unix epoch.

Return type *ClaimPredicate*

Returns a **before_absolute_time** claim predicate.

classmethod predicate_before_relative_time(*cls, seconds*)

Returns a **before_relative_time** claim predicate.

This predicate will be fulfilled if the closing time of the ledger that includes the *CreateClaimableBalance* operation plus this relative time delta (in seconds) is less than the current time.

Parameters **seconds** (*int*) – seconds since closeTime of the ledger in which the ClaimableBalanceEntry was created.

Return type *ClaimPredicate*

Returns a **before_relative_time** claim predicate.

classmethod predicate_not(*cls, predicate*)

Returns a **not** claim predicate.

Parameters **predicate** (*ClaimPredicate*) – a ClaimPredicate.

Return type *ClaimPredicate*

Returns a **not** claim predicate.

classmethod `predicate_or(cls, left, right)`

Returns an **or** claim predicate

Parameters

- **left** (*ClaimPredicate*) – a ClaimPredicate.
- **right** (*ClaimPredicate*) – a ClaimPredicate.

Return type *ClaimPredicate*

Returns an **or** claim predicate.

classmethod `predicate_unconditional(cls)`

Returns an **unconditional** claim predicate.

Return type *ClaimPredicate*

Returns an **unconditional** claim predicate.

class `stellar_sdk.operation.create_claimable_balance.ClaimPredicateType(value)`

Currently supported claim predicate types.

class `stellar_sdk.operation.create_claimable_balance.ClaimPredicateGroup(left, right)`

Used to assemble the left and right values for `and_predicates` and `or_predicates`.

Parameters

- **left** (*ClaimPredicate*) – The ClaimPredicate.
- **right** (*ClaimPredicate*) – The ClaimPredicate.

ClaimClaimableBalance

class `stellar_sdk.operation.ClaimClaimableBalance(balance_id, source=None)`

The *ClaimClaimableBalance* object, which represents a ClaimClaimableBalance operation on Stellar's network.

Claims a ClaimableBalanceEntry and adds the amount of asset on the entry to the source account.

Threshold: Low

See [Claim Claimable Balance](#) for more information.

Parameters

- **balance_id** (*str*) – The claimable balance id to be claimed.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

classmethod `from_xdr_object(cls, xdr_object)`

Creates a *ClaimClaimableBalance* object from an XDR Operation object.

Return type *ClaimClaimableBalance*

`to_xdr_object()`

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

BeginSponsoringFutureReserves

class stellar_sdk.operation.**BeginSponsoringFutureReserves**(*sponsored_id*, *source=None*)

The *BeginSponsoringFutureReserves* object, which represents a *BeginSponsoringFutureReserves* operation on Stellar's network.

Establishes the is-sponsoring-future-reserves-for relationship between the source account and sponsoredID. See [Sponsored Reserves](#) for more information.

Threshold: Medium

See [Begin Sponsoring Future Reserves](#) for more information.

Parameters

- **sponsored_id** (*str*) – The sponsored account id.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

classmethod **from_xdr_object**(*cls*, *xdr_object*)

Creates a *BeginSponsoringFutureReserves* object from an XDR Operation object.

Return type *BeginSponsoringFutureReserves*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

EndSponsoringFutureReserves

class stellar_sdk.operation.**EndSponsoringFutureReserves**(*source=None*)

The *EndSponsoringFutureReserves* object, which represents a *EndSponsoringFutureReserves* operation on Stellar's network.

Terminates the current is-sponsoring-future-reserves-for relationship in which the source account is sponsored. See [Sponsored Reserves](#) for more information.

Threshold: Medium

See [End Sponsoring Future Reserves](#).

Parameters **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

classmethod **from_xdr_object**(*cls*, *xdr_object*)

Creates a *EndSponsoringFutureReserves* object from an XDR Operation object.

Return type *EndSponsoringFutureReserves*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

RevokeSponsorship

```
class stellar_sdk.operation.RevokeSponsorship(revoke_sponsorship_type, account_id, trustline, offer,
                                             data, claimable_balance_id, signer, liquidity_pool_id,
                                             source=None)
```

The *RevokeSponsorship* object, which represents a RevokeSponsorship operation on Stellar’s network.

The logic of this operation depends on the state of the source account.

If the source account is not sponsored or is sponsored by the owner of the specified entry or sub-entry, then attempt to revoke the sponsorship. If the source account is sponsored, the next step depends on whether the entry is sponsored or not. If it is sponsored, attempt to transfer the sponsorship to the sponsor of the source account. If the entry is not sponsored, then establish the sponsorship. See [Sponsored Reserves](#) for more information.

Threshold: Medium

See [Revoke Sponsorship](#) for more information.

Parameters

- **revoke_sponsorship_type** (*RevokeSponsorshipType*) – The sponsored account id.
- **account_id** (*Optional[str]*) – The sponsored account ID.
- **trustline** (*Optional[TrustLine]*) – The sponsored trustline.
- **offer** (*Optional[Offer]*) – The sponsored offer.
- **data** (*Optional[Data]*) – The sponsored data.
- **claimable_balance_id** (*Optional[str]*) – The sponsored claimable balance.
- **signer** (*Optional[Signer]*) – The sponsored signer.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction’s source account.

```
classmethod from_xdr_object(cls, xdr_object)
```

Creates a *RevokeSponsorship* object from an XDR Operation object.

Return type *RevokeSponsorship*

```
to_xdr_object()
```

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

```
class stellar_sdk.operation.revoke_sponsorship.RevokeSponsorshipType(value)
```

Currently supported RevokeSponsorship types.

```
class stellar_sdk.operation.revoke_sponsorship.TrustLine(account_id, asset)
```

```
class stellar_sdk.operation.revoke_sponsorship.Offer(seller_id, offer_id)
```

```
class stellar_sdk.operation.revoke_sponsorship.Data(account_id, data_name)
```

```
class stellar_sdk.operation.revoke_sponsorship.Signer(account_id, signer_key)
```

Clawback

class stellar_sdk.operation.Clawback(*asset, from_, amount, source=None*)

The *Clawback* object, which represents a Clawback operation on Stellar's network.

Claws back an amount of an asset from an account.

Threshold: Medium

See [Clawback](#) for more information.

Parameters

- **asset** (*Asset*) – The asset being clawed back.
- **from** – The public key of the account to claw back from.
- **amount** (*Union[str, Decimal]*) – The amount of the asset to claw back.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

classmethod from_xdr_object(*cls, xdr_object*)

Creates a *Clawback* object from an XDR Operation object.

Return type *Clawback*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

ClawbackClaimableBalance

class stellar_sdk.operation.ClawbackClaimableBalance(*balance_id, source=None*)

The *ClawbackClaimableBalance* object, which represents a ClawbackClaimableBalance operation on Stellar's network.

Claws back a claimable balance

Threshold: Medium

See [Clawback Claimable Balance](#) for more information.

Parameters

- **balance_id** (*str*) – The claimable balance ID to be clawed back.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

classmethod from_xdr_object(*cls, xdr_object*)

Creates a *ClawbackClaimableBalance* object from an XDR Operation object.

Return type *ClawbackClaimableBalance*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

SetTrustLineFlags

class stellar_sdk.operation.SetTrustLineFlags(*trustor*, *asset*, *clear_flags=None*, *set_flags=None*, *source=None*)

The *SetTrustLineFlags* object, which represents a SetTrustLineFlags operation on Stellar's network.

Updates the flags of an existing trust line. This is called by the issuer of the related asset.

Threshold: Low

See [Set Trustline Flags](#) for more information.

Parameters

- **trustor** (*str*) – The account whose trustline this is.
- **asset** (*Asset*) – The asset on the trustline.
- **clear_flags** (*Optional[TrustLineFlags]*) – The flags to clear.
- **set_flags** (*Optional[TrustLineFlags]*) – The flags to set.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

classmethod from_xdr_object(*cls*, *xdr_object*)

Creates a *SetTrustLineFlags* object from an XDR Operation object.

Return type *SetTrustLineFlags*

to_xdr_object()

Creates an XDR Operation object that represents this *Operation*.

Return type *Operation*

class stellar_sdk.operation.set_trust_line_flags.TrustLineFlags(*value*)

Indicates which flags to set. For details about the flags, please refer to the [CAP-0035](#).

- **AUTHORIZED_FLAG**: issuer has authorized account to perform transactions with its credit
- **AUTHORIZED_TO_MAINTAIN_LIABILITIES_FLAG**: issuer has authorized account to maintain and reduce liabilities for its credit
- **TRUSTLINE_CLAWBACK_ENABLED_FLAG**: issuer has specified that it may clawback its credit, and that claimable balances created with its credit may also be clawed back

2.1.14 Price

class stellar_sdk.price.Price(*n*, *d*)

Create a new price. Price in Stellar is represented as a fraction.

An example:

```
from stellar_sdk import Price

price_a = Price(1, 2)
price_b = Price.from_raw_price("0.5")
```

Parameters

- **n** (*int*) – numerator
- **d** (*int*) – denominator

classmethod `from_raw_price(cls, price)`

Create a *Price* from the given str or Decimal price.

Parameters `price` (`Union[str, Decimal]`) – the str or Decimal price. (ex. "0.125")

Return type *Price*

Returns A new *Price* object from the given str or Decimal price.

Raises *NoApproximationError*: if the approximation could not not be found.

classmethod `from_xdr_object(cls, xdr_object)`

Create a *Price* from an XDR Price object.

Parameters `xdr_object` (*Price*) – The XDR Price object.

Return type *Price*

Returns A new *Price* object from the given XDR Price object.

to_xdr_object()

Returns the xdr object for this price object.

Return type *Price*

Returns XDR Price object

2.1.15 Server

class `stellar_sdk.server.Server`(`horizon_url='https://horizon-testnet.stellar.org'`, `client=None`)

Server handles the network connection to a [Horizon](#) instance and exposes an interface for requests to that instance.

An example:

```
from stellar_sdk import Server

server = Server("https://horizon-testnet.stellar.org")
resp = server.transactions().limit(10).order(desc=True).call()
print(resp)
```

Parameters

- **horizon_url** (`str`) – Horizon Server URL (ex. "https://horizon-testnet.stellar.org" for test network, "https://horizon-testnet.stellar.org" for public network)
- **client** (`Optional[BaseSyncClient]`) – Http client used to send the request

accounts()

Return type *AccountsCallBuilder*

Returns New `stellar_sdk.call_builder.call_builder_sync.AccountsCallBuilder` object configured by a current Horizon server configuration.

assets()

Return type *AssetsCallBuilder*

Returns New `stellar_sdk.call_builder.call_builder_sync.AssetsCallBuilder` object configured by a current Horizon server configuration.

claimable_balances()

Return type `ClaimableBalancesCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.ClaimableBalancesCallBuilder` object configured by a current Horizon server configuration.

close()

Close underlying connector, and release all acquired resources.

Return type `None`

data(account_id, data_name)

Return type `DataCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.DataCallBuilder` object configured by a current Horizon server configuration.

effects()

Return type `EffectsCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.EffectsCallBuilder` object configured by a current Horizon server configuration.

fee_stats()

Return type `FeeStatsCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.FeeStatsCallBuilder` object configured by a current Horizon server configuration.

fetch_base_fee()

Fetch the base fee. Since this hits the server, if the server call fails, you might get an error. You should be prepared to use a default value if that happens.

Return type `int`

Returns the base fee

Raises `ConnectionError` `NotFoundError` `BadRequestError` `BadResponseError` `UnknownRequestError`

ledgers()

Return type `LedgersCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.LedgersCallBuilder` object configured by a current Horizon server configuration.

liquidity_pools()

Return type `LiquidityPoolsBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.LiquidityPoolsBuilder` object configured by a current Horizon server configuration.

load_account(*account_id*)

Fetches an account's most current base state (like sequence) in the ledger and then creates and returns an `stellar_sdk.account.Account` object.

If you want to get complete account information, please use `stellar_sdk.server.Server.accounts()`.

Parameters **account_id** (Union[`MuxedAccount`, `Keypair`, `str`]) – The account to load.

Return type `Account`

Returns an `stellar_sdk.account.Account` object.

Raises `ConnectionError` `NotFoundError` `BadRequestError` `BadResponseError` `UnknownRequestError`

offers()

Return type `OffersCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.OffersCallBuilder` object configured by a current Horizon server configuration.

operations()

Return type `OperationsCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.OperationsCallBuilder` object configured by a current Horizon server configuration.

orderbook(*selling*, *buying*)

Parameters

- **selling** (`Asset`) – Asset being sold
- **buying** (`Asset`) – Asset being bought

Return type `OrderbookCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.OrderbookCallBuilder` object configured by a current Horizon server configuration.

payments()

Return type `PaymentsCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.PaymentsCallBuilder` object configured by a current Horizon server configuration.

root()

Return type `RootCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.RootCallBuilder` object configured by a current Horizon server configuration.

strict_receive_paths(*source*, *destination_asset*, *destination_amount*)

Parameters

- **source** (`Union[str, List[Asset]]`) – The sender’s account ID or a list of Assets. Any returned path must use a source that the sender can hold.
- **destination_asset** (`Asset`) – The destination asset.
- **destination_amount** (`str`) – The amount, denominated in the destination asset, that any returned path should be able to satisfy.

Return type `StrictReceivePathsCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.StrictReceivePathsCallBuilder` object configured by a current Horizon server configuration.

strict_send_paths(*source_asset*, *source_amount*, *destination*)

Parameters

- **source_asset** (`Asset`) – The asset to be sent.
- **source_amount** (`str`) – The amount, denominated in the source asset, that any returned path should be able to satisfy.
- **destination** (`Union[str, List[Asset]]`) – The destination account or the destination assets.

Return type `StrictSendPathsCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.StrictReceivePathsCallBuilder` object configured by a current Horizon server configuration.

submit_transaction(*transaction_envelope*, *skip_memo_required_check=False*)

Submits a transaction to the network.

Parameters

- **transaction_envelope** (`Union[TransactionEnvelope, FeeBumpTransactionEnvelope, str]`) – `stellar_sdk.transaction_envelope.TransactionEnvelope` object or base64 encoded xdr
- **skip_memo_required_check** (`bool`) – Allow skipping memo

Return type `Dict[str, Any]`

Returns the response from horizon

Raises `ConnectionError` `NotFoundError` `BadRequestError` `BadResponseError` `UnknownRequestError` `AccountRequiresMemoError`

trade_aggregations(*base*, *counter*, *resolution*, *start_time=None*, *end_time=None*, *offset=None*)

Parameters

- **base** (`Asset`) – base asset
- **counter** (`Asset`) – counter asset

- **resolution** (`int`) – segment duration as millis since epoch. *Supported values are 1 minute (60000), 5 minutes (300000), 15 minutes (900000), 1 hour (3600000), 1 day (86400000) and 1 week (604800000).*
- **start_time** (`Optional[int]`) – lower time boundary represented as millis since epoch
- **end_time** (`Optional[int]`) – upper time boundary represented as millis since epoch
- **offset** (`Optional[int]`) – segments can be offset using this parameter. Expressed in milliseconds. *Can only be used if the resolution is greater than 1 hour. Value must be in whole hours, less than the provided resolution, and less than 24 hours.*

Return type `TradeAggregationsCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.TradeAggregationsCallBuilder` object configured by a current Horizon server configuration.

`trades()`

Return type `TradesCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.TradesCallBuilder` object configured by a current Horizon server configuration.

`transactions()`

Return type `TransactionsCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_sync.TransactionsCallBuilder` object configured by a current Horizon server configuration.

2.1.16 ServerAsync

```
class stellar_sdk.server_async.ServerAsync(horizon_url='https://horizon-testnet.stellar.org',
                                           client=None)
```

ServerAsync handles the network connection to a [Horizon](#) instance and exposes an interface for requests to that instance.

An example:

```
import asyncio
from stellar_sdk import ServerAsync

async def example():
    async with ServerAsync("https://horizon-testnet.stellar.org") as server:
        resp = await server.transactions().limit(10).order(desc=True).call()
        print(resp)

asyncio.run(example())
```

Parameters

- **horizon_url** (`str`) – Horizon Server URL (ex. "https://horizon-testnet.stellar.org" for test network, "https://horizon-testnet.stellar.org" for public network)
- **client** (`Optional[BaseAsyncClient]`) – Http client used to send the request

accounts()

Return type *AccountsCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.AccountsCallBuilder* object configured by a current Horizon server configuration.

assets()

Return type *AssetsCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.AssetsCallBuilder* object configured by a current Horizon server configuration.

claimable_balances()

Return type *ClaimableBalancesCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.ClaimableBalancesCallBuilder* object configured by a current Horizon server configuration.

async close()

Close underlying connector, and release all acquired resources.

Return type *None*

data(*account_id*, *data_name*)

Return type *DataCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.DataCallBuilder* object configured by a current Horizon server configuration.

effects()

Return type *EffectsCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.EffectsCallBuilder* object configured by a current Horizon server configuration.

fee_stats()

Return type *FeeStatsCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.FeeStatsCallBuilder* object configured by a current Horizon server configuration.

async fetch_base_fee()

Fetch the base fee. Since this hits the server, if the server call fails, you might get an error. You should be prepared to use a default value if that happens.

Return type *int*

Returns the base fee

Raises *ConnectionError* *NotFoundError* *BadRequestError* *BadResponseError* *UnknownRequestError*

ledgers()

Return type *LedgersCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.LedgersCallBuilder* object configured by a current Horizon server configuration.

liquidity_pools()

Return type *LiquidityPoolsBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.LiquidityPoolsBuilder* object configured by a current Horizon server configuration.

async load_account(account_id)

Fetches an account's most current base state (like sequence) in the ledger and then creates and returns an *stellar_sdk.account.Account* object.

If you want to get complete account information, please use *stellar_sdk.server.Server.accounts()*.

Parameters **account_id** (*Union[MuxedAccount, Keypair, str]*) – The account to load.

Return type *Account*

Returns an *stellar_sdk.account.Account* object.

Raises *ConnectionError* *NotFoundError* *BadRequestError* *BadResponseError* *UnknownRequestError*

offers()

Return type *OffersCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.OffersCallBuilder* object configured by a current Horizon server configuration.

operations()

Return type *OperationsCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder* object configured by a current Horizon server configuration.

orderbook(selling, buying)

Parameters

- **selling** (*Asset*) – Asset being sold
- **buying** (*Asset*) – Asset being bought

Return type *OrderbookCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.OrderbookCallBuilder* object configured by a current Horizon server configuration.

payments()

Return type *PaymentsCallBuilder*

Returns New `stellar_sdk.call_builder.call_builder_async.PaymentsCallBuilder` object configured by a current Horizon server configuration.

`root()`

Return type `RootCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_async.RootCallBuilder` object configured by a current Horizon server configuration.

`strict_receive_paths(source, destination_asset, destination_amount)`

Parameters

- **source** (`Union[str, List[Asset]]`) – The sender’s account ID or a list of Assets. Any returned path must use a source that the sender can hold.
- **destination_asset** (`Asset`) – The destination asset.
- **destination_amount** (`str`) – The amount, denominated in the destination asset, that any returned path should be able to satisfy.

Return type `StrictReceivePathsCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_async.StrictReceivePathsCallBuilder` object configured by a current Horizon server configuration.

`strict_send_paths(source_asset, source_amount, destination)`

Parameters

- **source_asset** (`Asset`) – The asset to be sent.
- **source_amount** (`str`) – The amount, denominated in the source asset, that any returned path should be able to satisfy.
- **destination** (`Union[str, List[Asset]]`) – The destination account or the destination assets.

Return type `StrictSendPathsCallBuilder`

Returns New `stellar_sdk.call_builder.call_builder_async.StrictReceivePathsCallBuilder` object configured by a current Horizon server configuration.

`async submit_transaction(transaction_envelope, skip_memo_required_check=False)`

Submits a transaction to the network.

Parameters

- **transaction_envelope** (`Union[TransactionEnvelope, FeeBumpTransactionEnvelope, str]`) – `stellar_sdk.transaction_envelope.TransactionEnvelope` object or base64 encoded xdr
- **skip_memo_required_check** (`bool`) – Allow skipping memo

Return type `Dict[str, Any]`

Returns the response from horizon

Raises `ConnectionError` `NotFoundError` `BadRequestError` `BadResponseError` `UnknownRequestError` `AccountRequiresMemoError`

trade_aggregations(*base, counter, resolution, start_time=None, end_time=None, offset=None*)

Parameters

- **base** (*Asset*) – base asset
- **counter** (*Asset*) – counter asset
- **resolution** (*int*) – segment duration as millis since epoch. *Supported values are 1 minute (60000), 5 minutes (300000), 15 minutes (900000), 1 hour (3600000), 1 day (86400000) and 1 week (604800000).*
- **start_time** (*Optional[int]*) – lower time boundary represented as millis since epoch
- **end_time** (*Optional[int]*) – upper time boundary represented as millis since epoch
- **offset** (*Optional[int]*) – segments can be offset using this parameter. Expressed in milliseconds. *Can only be used if the resolution is greater than 1 hour. Value must be in whole hours, less than the provided resolution, and less than 24 hours.*

Return type *TradeAggregationsCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.TradeAggregationsCallBuilder* object configured by a current Horizon server configuration.

trades()

Return type *TradesCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.TradesCallBuilder* object configured by a current Horizon server configuration.

transactions()

Return type *TransactionsCallBuilder*

Returns New *stellar_sdk.call_builder.call_builder_async.TransactionsCallBuilder* object configured by a current Horizon server configuration.

2.1.17 Signer

class *stellar_sdk.signer.Signer*(*signer_key, weight*)

The *Signer* object, which represents an account signer on Stellar’s network.

An example:

```
from stellar_sdk import Signer

signer_ed25519 = Signer.ed25519_public_key("GCC3U63F5OJIG4VS6XCFUJGCQRRMNCV GASDGIZZ EPA3AZ242K4J
1) signer_sha256_hash = Signer.sha256_hash("XCC3U63F5OJIG4VS6XCFUJGCQRRMNCV GASDGIZZ EPA3AZ242K4J
2) signer_pre_auth_tx = Signer.pre_auth_tx("TCC3U63F5OJIG4VS6XCFUJGCQRRMNCV GASDGIZZ EPA3AZ242K4J
3) print(f'signer_ed25519 account id: {signer_ed25519.signer_key.encoded_signer_key}')
print(f'signer_ed25519 weight: {signer_ed25519.weight}')
```

Parameters

- **signer_key** (*SignerKey*) – The signer object
- **weight** (*int*) – The weight of the key

classmethod `ed25519_public_key`(*cls, account_id, weight*)

Create ED25519 PUBLIC KEY Signer from account id.

Parameters

- **account_id** (`Union[str, bytes]`) – account id (ex. "GDNA2V62PVEFBZ74CDJKTUHLY4Y7PL5UAV2MAM4VWF6USFE3SH2354AD")
- **weight** (`int`) – The weight of the signer (0 to delete or 1-255)

Return type *Signer*

Returns ED25519 PUBLIC KEY Signer

Raises *Ed25519PublicKeyInvalidError*: if `account_id` is not a valid ed25519 public key.

classmethod `from_xdr_object`(*cls, xdr_object*)

Create a *Signer* from an XDR Signer object.

Parameters `xdr_object` (*Signer*) – The XDR Signer object.

Return type *Signer*

Returns A new *Signer* object from the given XDR Signer object.

classmethod `pre_auth_tx`(*cls, pre_auth_tx_hash, weight*)

Create Pre AUTH TX Signer from the sha256 hash of a transaction, click [here](#) for more information.

Parameters

- **pre_auth_tx_hash** (`Union[str, bytes]`) – The sha256 hash of a transaction (ex. "TDNA2V62PVEFBZ74CDJKTUHLY4Y7PL5UAV2MAM4VWF6USFE3SH234BSS" or bytes)
- **weight** (`int`) – The weight of the signer (0 to delete or 1-255)

Return type *Signer*

Returns Pre AUTH TX Signer

classmethod `sha256_hash`(*cls, sha256_hash, weight*)

Create SHA256 HASH Signer from a sha256 hash of a preimage, click [here](#) for more information.

Parameters

- **sha256_hash** (`Union[str, bytes]`) – a sha256 hash of a preimage (ex. "XDNA2V62PVEFBZ74CDJKTUHLY4Y7PL5UAV2MAM4VWF6USFE3SH235FXL" or bytes)
- **weight** (`int`) – The weight of the signer (0 to delete or 1-255)

Return type *Signer*

Returns SHA256 HASH Signer

to_xdr_object()

Returns the xdr object for this Signer object.

Return type *Signer*

Returns XDR Signer object

2.1.18 SignerKey

class stellar_sdk.signer_key.**SignerKey**(*signer_key, signer_key_type*)

The *SignerKey* object, which represents an account signer key on Stellar's network.

Parameters

- **signer_key** (*bytes*) – The signer key.
- **signer_key** – The signer key type.

classmethod **ed25519_public_key**(*cls, account_id*)

Create ED25519 PUBLIC KEY Signer from account id.

Parameters **account_id** (*Union[str, bytes]*) – account id

Return type *SignerKey*

Returns ED25519 PUBLIC KEY Signer

Raises *Ed25519PublicKeyInvalidError*: if *account_id* is not a valid ed25519 public key.

property **encoded_signer_key**: **str**

return: The signer key encoded in Strkey format.

Return type *str*

classmethod **from_xdr_object**(*cls, xdr_object*)

Create a *SignerKey* from an XDR SignerKey object.

Parameters **xdr_object** (*SignerKey*) – The XDR SignerKey object.

Return type *SignerKey*

Returns A new *SignerKey* object from the given XDR SignerKey object.

classmethod **pre_auth_tx**(*cls, pre_auth_tx_hash*)

Create Pre AUTH TX Signer from the sha256 hash of a transaction, click [here](#) for more information.

Parameters **pre_auth_tx_hash** (*Union[str, bytes]*) – The sha256 hash of a transaction.

Return type *SignerKey*

Returns Pre AUTH TX Signer

classmethod **sha256_hash**(*cls, sha256_hash*)

Create SHA256 HASH Signer from a sha256 hash of a preimage, click [here](#) for more information.

Parameters **sha256_hash** (*Union[str, bytes]*) – a sha256 hash of a preimage

Return type *SignerKey*

Returns SHA256 HASH Signer

to_xdr_object()

Returns the xdr object for this SignerKey object.

Return type *SignerKey*

Returns XDR Signer object

class stellar_sdk.signer_key.**SignerKeyType**(*value*)

An enumeration.

2.1.19 StrKey

class stellar_sdk.strkey.StrKey

StrKey is a helper class that allows encoding and decoding strkey.

static decode_ed25519_public_key(*data*)

Decodes encoded ed25519 public key strkey to raw data.

Parameters *data* (*str*) – encoded ed25519 public key strkey

Return type *bytes*

Returns raw bytes

Raises *Ed25519PublicKeyInvalidError*

static decode_ed25519_secret_seed(*data*)

Decodes encoded ed25519 secret seed strkey to raw data.

Parameters *data* (*str*) – encoded ed25519 secret seed strkey

Return type *bytes*

Returns raw bytes

Raises *Ed25519SecretSeedInvalidError*

static decode_muxed_account(*data*)

Decodes encoded muxed account strkey to raw data.

Parameters *data* (*str*) – encoded muxed account strkey

Return type *MuxedAccount*

Returns raw bytes

Raises *ValueError*

static decode_pre_auth_tx(*data*)

Decodes encoded pre auth tx strkey to raw data.

Parameters *data* (*str*) – encoded pre auth tx strkey

Return type *bytes*

Returns raw bytes

Raises *ValueError*

static decode_sha256_hash(*data*)

Decodes encoded sha256 hash strkey to raw data.

Parameters *data* (*str*) – encoded sha256 hash strkey

Return type *bytes*

Returns raw bytes

Raises *ValueError*

static encode_ed25519_public_key(*data*)

Encodes data to encoded ed25519 public key strkey.

Parameters *data* (*bytes*) – data to encode

Return type *str*

Returns encoded ed25519 public key strkey

Raises ValueError

static `encode_ed25519_secret_seed(data)`

Encodes data to encoded ed25519 secret seed strkey.

Parameters `data` (`bytes`) – data to encode

Return type `str`

Returns encoded ed25519 secret seed strkey

Raises ValueError

static `encode_muxed_account(data)`

Encodes data to encoded muxed account strkey.

Parameters `data` (`MuxedAccount`) – data to encode

Return type `str`

Returns encoded muxed account strkey

Raises ValueError

static `encode_pre_auth_tx(data)`

Encodes data to encoded pre auth tx strkey.

Parameters `data` (`bytes`) – data to encode

Return type `str`

Returns encoded pre auth tx strkey

Raises ValueError

static `encode_sha256_hash(data)`

Encodes data to encoded sha256 hash strkey.

Parameters `data` (`bytes`) – data to encode

Return type `str`

Returns encoded sha256 hash strkey

Raises ValueError

static `is_valid_ed25519_public_key(public_key)`

Returns True if the given `seed` is a valid ed25519 public key strkey.

Parameters `public_key` (`str`) – encoded ed25519 public key strkey

Return type `bool`

Returns True if the given key is valid

static `is_valid_ed25519_secret_seed(seed)`

Returns True if the given `seed` is a valid ed25519 secret seed strkey.

Parameters `seed` (`str`) – encoded ed25519 secret seed strkey

Return type `bool`

Returns True if the given key is valid

static `is_valid_pre_auth_tx(pre_auth_tx)`

Returns True if the given `pre_auth_tx` is a valid encoded pre auth tx strkey.

Parameters `pre_auth_tx` (`str`) – encoded pre auth tx strkey

Return type `bool`

Returns True if the given key is valid

static is_valid_sha256_hash(*sha256_hash*)

Returns True if the given *sha256_hash* is a valid encoded sha256 hash(HashX) strkey.

Parameters *sha256_hash* (`str`) – encoded sha256 hash(HashX) strkey

Return type `bool`

Returns True if the given key is valid

2.1.20 TimeBounds

class `stellar_sdk.time_bounds.TimeBounds`(*min_time*, *max_time*)

`TimeBounds` represents the time interval that a transaction is valid.

The UNIX timestamp (in seconds), determined by ledger time, of a lower and upper bound of when this transaction will be valid. If a transaction is submitted too early or too late, it will fail to make it into the transaction set. *max_time* equal 0 means that it's not set.

See [Stellar's documentation on Transactions](#) for more information on how `TimeBounds` are used within transactions.

Parameters

- **min_time** (`int`) – the UNIX timestamp (in seconds)
- **max_time** (`int`) – the UNIX timestamp (in seconds)

Raises `ValueError`: if *max_time* less than *min_time*.

classmethod `from_xdr_object`(*cls*, *xdr_object*)

Create a `TimeBounds` from an XDR `TimeBounds` object.

Parameters *xdr_object* (`TimeBounds`) – The XDR `TimeBounds` object.

Return type `TimeBounds`

Returns A new `TimeBounds` object from the given XDR `TimeBounds` object.

to_xdr_object()

Returns the xdr object for this `TimeBounds` object.

Return type `TimeBounds`

Returns XDR `TimeBounds` object

2.1.21 DecoratedSignature

class `stellar_sdk.decorated_signature.DecoratedSignature`(*signature_hint*, *signature*)

classmethod `from_xdr_object`(*cls*, *xdr_object*)

Create a `DecoratedSignature` from an XDR `DecoratedSignature` object.

Parameters *xdr_object* (`DecoratedSignature`) – The XDR `DecoratedSignature` object.

Return type `DecoratedSignature`

Returns A new `DecoratedSignature` object from the given XDR `DecoratedSignature` object.

`to_xdr_object()`

Returns the xdr object for this DecoratedSignature object.

Return type *DecoratedSignature*

Returns XDR DecoratedSignature object

2.1.22 Transaction

class `stellar_sdk.transaction.Transaction`(*source, sequence, fee, operations, memo=None, time_bounds=None, v1=True*)

The *Transaction* object, which represents a transaction (Transaction or TransactionV0) on Stellar's network.

A transaction contains a list of operations, which are all executed in order as one ACID transaction, along with an associated source account, fee, account sequence number, list of signatures, both an optional memo and an optional TimeBounds. Typically a *Transaction* is placed in a *TransactionEnvelope* which is then signed before being sent over the network.

For more information on Transactions in Stellar, see [Stellar's guide on transactions](#).

Parameters

- **source** (`Union[MixedAccount, Keypair, str]`) – the source account for the transaction.
- **sequence** (`int`) – The sequence number for the transaction.
- **fee** (`int`) – The max fee amount for the transaction, which should equal FEE (currently least 100 stroops) multiplied by the number of operations in the transaction. See [Stellar's latest documentation on fees](#) for more information.
- **operations** (`List[Operation]`) – A list of operations objects (typically its subclasses as defined in `stellar_sdk.operation.Operation`).
- **time_bounds** (`Optional[TimeBounds]`) – The timebounds for the validity of this transaction.
- **memo** (`Optional[Memo]`) – The memo being sent with the transaction, being represented as one of the subclasses of the *Memo* object.
- **v1** (`bool`) – When this value is set to `True`, V1 transactions will be generated, otherwise V0 transactions will be generated. See [CAP-0015](#) for more information.

classmethod `from_xdr`(*cls, xdr, v1=True*)

Create a new *Transaction* from an XDR string.

Parameters

- **xdr** (`str`) – The XDR string that represents a transaction.
- **v1** (`bool`) – Temporary feature flag to allow alpha testing of Stellar Protocol 13 transactions. We will remove this once all transactions are supposed to be v1. See [CAP-0015](#) for more information.

Return type *Transaction*

Returns A new *Transaction* object from the given XDR Transaction base64 string object.

classmethod `from_xdr_object`(*cls, xdr_object, v1=True*)

Create a new *Transaction* from an XDR object.

Parameters

- **xdr_object** (`Union[Transaction, TransactionV0]`) – The XDR object that represents a transaction.

- **v1** (`bool`) – Temporary feature flag to allow alpha testing of Stellar Protocol 13 transactions. We will remove this once all transactions are supposed to be v1. See [CAP-0015](#) for more information.

Return type `Transaction`

Returns A new `Transaction` object from the given XDR Transaction object.

get_claimable_balance_id(`operation_index`)

Calculate the claimable balance ID for an operation within the transaction.

Parameters `operation_index` (`int`) – the index of the CreateClaimableBalance operation.

Return type `str`

Returns a hex string representing the claimable balance ID.

Raises

`IndexError`: if `operation_index` is invalid.

`TypeError`: if operation at `operation_index` is not `FeeBumpTransactionEnvelope`.

to_xdr_object()

Get an XDR object representation of this `Transaction`.

Return type `Union[Transaction, TransactionV0]`

Returns XDR Transaction object

2.1.23 TransactionEnvelope

class `stellar_sdk.transaction_envelope.TransactionEnvelope`(`transaction`, `network_passphrase`, `signatures=None`)

The `TransactionEnvelope` object, which represents a transaction envelope ready to sign and submit to send over the network.

When a transaction is ready to be prepared for sending over the network, it must be put into a `TransactionEnvelope`, which includes additional metadata such as the signers for a given transaction. Ultimately, this class handles signing and conversion to and from XDR for usage on Stellar's network.

Parameters

- **transaction** (`Transaction`) – The transaction that is encapsulated in this envelope.
- **signatures** (`list`) – which contains a list of signatures that have already been created.
- **network_passphrase** (`str`) – The network to connect to for verifying and retrieving additional attributes from.

classmethod `from_xdr`(`cls`, `xdr`, `network_passphrase`)

Create a new `BaseTransactionEnvelope` from an XDR string.

Parameters

- **xdr** (`str`) – The XDR string that represents a transaction envelope.
- **network_passphrase** (`str`) – which network this transaction envelope is associated with.

Return type `~T`

Returns A new `BaseTransactionEnvelope` object from the given XDR `TransactionEnvelope` base64 string object.

classmethod `from_xdr_object(cls, xdr_object, network_passphrase)`

Create a new *TransactionEnvelope* from an XDR object.

Parameters

- **xdr_object** (*TransactionEnvelope*) – The XDR object that represents a transaction envelope.
- **network_passphrase** (*str*) – The network to connect to for verifying and retrieving additional attributes from.

Return type *TransactionEnvelope*

Returns A new *TransactionEnvelope* object from the given XDR *TransactionEnvelope* object.

hash()

Get the XDR Hash of the signature base.

This hash is ultimately what is signed before transactions are sent over the network. See *signature_base()* for more details about this process.

Return type *bytes*

Returns The XDR Hash of this transaction envelope’s signature base.

hash_hex()

Return a hex encoded hash for this transaction envelope.

Return type *str*

Returns A hex encoded hash for this transaction envelope.

sign(signer)

Sign this transaction envelope with a given keypair.

Note that the signature must not already be in this instance’s list of signatures.

Parameters **signer** (*Union*[*Keypair*, *str*]) – The keypair or secret to use for signing this transaction envelope.

Raise *SignatureExistError*: if this signature already exists.

Return type *None*

sign_hashx(preimage)

Sign this transaction envelope with a Hash(x) signature.

See Stellar’s documentation on *Multi-Sig* for more details on Hash(x) signatures.

Parameters **preimage** (*Union*[*bytes*, *str*]) – Preimage of hash used as signer, byte hash or hex encoded string

Return type *None*

signature_base()

Get the signature base of this transaction envelope.

Return the “signature base” of this transaction, which is the value that, when hashed, should be signed to create a signature that validators on the Stellar Network will accept.

It is composed of a 4 prefix bytes followed by the xdr-encoded form of this transaction.

Return type *bytes*

Returns The signature base of this transaction envelope.

to_transaction_envelope_v1()

Create a new *TransactionEnvelope*, if the internal tx is not v1, we will convert it to v1.

Return type *TransactionEnvelope*

to_xdr()

Get the base64 encoded XDR string representing this BaseTransactionEnvelope.

Return type *str*

Returns XDR TransactionEnvelope base64 string object

to_xdr_object()

Get an XDR object representation of this *TransactionEnvelope*.

Return type *TransactionEnvelope*

Returns XDR TransactionEnvelope object

2.1.24 FeeBumpTransaction

class stellar_sdk.fee_bump_transaction.**FeeBumpTransaction**(*fee_source, base_fee, inner_transaction_envelope*)

The *FeeBumpTransaction* object, which represents a fee bump transaction on Stellar's network.

See [Fee-Bump Transactions](#) for more information. See [CAP-0015](#) for more information.

Parameters

- **fee_source** (*Union[MixedAccount, Keypair, str]*) – The account paying for the transaction.
- **base_fee** (*int*) – The max fee willing to pay per operation in inner transaction (**in stroops**).
- **inner_transaction_envelope** (*TransactionEnvelope*) – The TransactionEnvelope to be bumped by the fee bump transaction.

classmethod **from_xdr**(*cls, xdr, network_passphrase*)

Create a new *FeeBumpTransaction* from an XDR string.

Parameters

- **xdr** (*str*) – The XDR string that represents a transaction.
- **network_passphrase** (*str*) – The network to connect to for verifying and retrieving additional attributes from.

Return type *FeeBumpTransaction*

Returns A new *FeeBumpTransaction* object from the given XDR FeeBumpTransaction base64 string object.

classmethod **from_xdr_object**(*cls, xdr_object, network_passphrase*)

Create a new *FeeBumpTransaction* from an XDR object.

Parameters

- **xdr_object** (*FeeBumpTransaction*) – The XDR object that represents a fee bump transaction.
- **network_passphrase** (*str*) – The network to connect to for verifying and retrieving additional attributes from.

Return type *FeeBumpTransaction*

Returns A new *FeeBumpTransaction* object from the given XDR Transaction object.

to_xdr_object()

Get an XDR object representation of this *FeeBumpTransaction*.

Return type *FeeBumpTransaction*

Returns XDR Transaction object

2.1.25 FeeBumpTransactionEnvelope

```
class stellar_sdk.fee_bump_transaction_envelope.FeeBumpTransactionEnvelope(transaction, network_passphrase, signatures=None)
```

The *FeeBumpTransactionEnvelope* object, which represents a fee bump transaction envelope ready to sign and submit to send over the network.

When a fee bump transaction is ready to be prepared for sending over the network, it must be put into a *FeeBumpTransactionEnvelope*, which includes additional metadata such as the signers for a given transaction. Ultimately, this class handles signing and conversion to and from XDR for usage on Stellar's network.

See [Fee-Bump Transactions](#) for more information. See [CAP-0015](#) for more information.

Parameters

- **transaction** (*FeeBumpTransaction*) – The fee bump transaction that is encapsulated in this envelope.
- **signatures** (*list*) – which contains a list of signatures that have already been created.
- **network_passphrase** (*str*) – The network to connect to for verifying and retrieving additional attributes from.

```
classmethod from_xdr(cls, xdr, network_passphrase)
```

Create a new *BaseTransactionEnvelope* from an XDR string.

Parameters

- **xdr** (*str*) – The XDR string that represents a transaction envelope.
- **network_passphrase** (*str*) – which network this transaction envelope is associated with.

Return type ~T

Returns A new *BaseTransactionEnvelope* object from the given XDR *TransactionEnvelope* base64 string object.

```
classmethod from_xdr_object(cls, xdr_object, network_passphrase)
```

Create a new *FeeBumpTransactionEnvelope* from an XDR object.

Parameters

- **xdr_object** (*TransactionEnvelope*) – The XDR object that represents a fee bump transaction envelope.
- **network_passphrase** (*str*) – The network to connect to for verifying and retrieving additional attributes from.

Return type *FeeBumpTransactionEnvelope*

Returns A new *FeeBumpTransactionEnvelope* object from the given XDR *TransactionEnvelope* object.

hash()

Get the XDR Hash of the signature base.

This hash is ultimately what is signed before transactions are sent over the network. See [signature_base\(\)](#) for more details about this process.

Return type `bytes`

Returns The XDR Hash of this transaction envelope’s signature base.

hash_hex()

Return a hex encoded hash for this transaction envelope.

Return type `str`

Returns A hex encoded hash for this transaction envelope.

sign(*signer*)

Sign this transaction envelope with a given keypair.

Note that the signature must not already be in this instance’s list of signatures.

Parameters **signer** (`Union[Keypair, str]`) – The keypair or secret to use for signing this transaction envelope.

Raise `SignatureExistError`: if this signature already exists.

Return type `None`

sign_hashx(*preimage*)

Sign this transaction envelope with a Hash(x) signature.

See Stellar’s documentation on [Multi-Sig](#) for more details on Hash(x) signatures.

Parameters **preimage** (`Union[bytes, str]`) – Preimage of hash used as signer, byte hash or hex encoded string

Return type `None`

signature_base()

Get the signature base of this transaction envelope.

Return the “signature base” of this transaction, which is the value that, when hashed, should be signed to create a signature that validators on the Stellar Network will accept.

It is composed of a 4 prefix bytes followed by the xdr-encoded form of this transaction.

Return type `bytes`

Returns The signature base of this transaction envelope.

to_xdr()

Get the base64 encoded XDR string representing this `BaseTransactionEnvelope`.

Return type `str`

Returns XDR `TransactionEnvelope` base64 string object

to_xdr_object()

Get an XDR object representation of this `TransactionEnvelope`.

Return type `TransactionEnvelope`

Returns XDR `TransactionEnvelope` object

2.1.26 TransactionBuilder

```
class stellar_sdk.transaction_builder.TransactionBuilder(source_account,
                                                         network_passphrase='Test SDF Network ;
                                                         September 2015', base_fee=100,
                                                         v1=True)
```

Transaction builder helps constructs a new *TransactionEnvelope* using the given *Account* as the transaction's "source account". The transaction will use the current sequence number of the given account as its sequence number and increment the given account's sequence number by one. The given source account must include a private key for signing the transaction or an error will be thrown.

Be careful about **unsubmitted transactions**! When you build a transaction, stellar-sdk automatically increments the source account's sequence number. If you end up not submitting this transaction and submitting another one instead, it'll fail due to the sequence number being wrong. So if you decide not to use a built transaction, make sure to update the source account's sequence number with *stellar_sdk.server.Server.load_account()* or *stellar_sdk.server_async.ServerAsync.load_account()* before creating another transaction.

An example:

```
# Alice pay 10.25 XLM to Bob
from stellar_sdk import Server, Asset, Keypair, TransactionBuilder, Network

alice_keypair = Keypair.from_secret(
    ↪"SBFZCHU5645DOKRWYBXVOXY2ELGJKFRX6VGGPRYUWHQ7PMXXJNDZFMKD")
bob_address = "GA7YNBW5CBTJZ3ZZOWX3ZNBKD6OE7A7IHUQVWVY62W2ZBG2SGZVOOPVH"

server = Server("https://horizon-testnet.stellar.org")
alice_account = server.load_account(alice_keypair.public_key)
network_passphrase = Network.TESTNET_NETWORK_PASSPHRASE
base_fee = 100
transaction = (
    TransactionBuilder(
        source_account=alice_account,
        network_passphrase=network_passphrase,
        base_fee=base_fee,
    )
    .add_text_memo("Hello, Stellar!")
    .append_payment_op(bob_address, Asset.native(), "10.25")
    .set_timeout(30)
    .build()
)
transaction.sign(alice_keypair)
response = server.submit_transaction(transaction)
print(response)
```

Parameters

- **source_account** (*Account*) – The source account for this transaction.
- **network_passphrase** (*str*) – The network to connect to for verifying and retrieving additional attributes from. Defaults to Test SDF Network ; September 2015.
- **base_fee** (*int*) – Max fee you're willing to pay per operation in this transaction (**in stroops**).
- **v1** (*bool*) – When this value is set to True, V1 transactions will be generated, otherwise V0 transactions will be generated. See [CAP-0015](#) for more information.

add_hash_memo(*memo_hash*)

Set the memo for the transaction to a new *HashMemo*.

Parameters **memo_hash** (`Union[bytes, str]`) – A 32 byte hash or hex encoded string to use as the memo.

Return type *TransactionBuilder*

Returns This builder instance.

Raises *MemoInvalidException*: if *memo_hash* is not a valid hash memo.

add_id_memo(*memo_id*)

Set the memo for the transaction to a new *IdMemo*.

Parameters **memo_id** (`int`) – A 64 bit unsigned integer to set as the memo.

Return type *TransactionBuilder*

Returns This builder instance.

Raises *MemoInvalidException*: if *memo_id* is not a valid id memo.

add_memo(*memo*)

Set the memo for the transaction build by this Builder.

Parameters **memo** (*Memo*) – A memo to add to this transaction.

Return type *TransactionBuilder*

Returns This builder instance.

add_return_hash_memo(*memo_return*)

Set the memo for the transaction to a new *RetHashMemo*.

Parameters **memo_return** (`Union[bytes, str]`) – A 32 byte hash or hex encoded string intended to be interpreted as the hash of the transaction the sender is refunding.

Return type *TransactionBuilder*

Returns This builder instance.

Raises *MemoInvalidException*: if *memo_return* is not a valid return hash memo.

add_text_memo(*memo_text*)

Set the memo for the transaction to a new *TextMemo*.

Parameters **memo_text** (`Union[str, bytes]`) – The text for the memo to add.

Return type *TransactionBuilder*

Returns This builder instance.

Raises *MemoInvalidException*: if *memo_text* is not a valid text memo.

add_time_bounds(*min_time*, *max_time*)

Add a time bound to this transaction.

Add a UNIX timestamp, determined by ledger time, of a lower and upper bound of when this transaction will be valid. If a transaction is submitted too early or too late, it will fail to make it into the transaction set. *max_time* equal 0 means that it's not set.

Parameters

- **min_time** (`int`) – the UNIX timestamp (in seconds)
- **max_time** (`int`) – the UNIX timestamp (in seconds)

Return type *TransactionBuilder*

Returns This builder instance.

append_account_merge_op(*destination, source=None*)

Append a *AccountMerge* operation to the list of operations.

Parameters

- **destination** (*Union[MixedAccount, str]*) – The ID of the offer. 0 for new offer. Set to existing offer ID to update or delete.
- **source** (*Union[MixedAccount, str, None]*) – The source address that is being merged into the destination account.

Return type *TransactionBuilder*

Returns This builder instance.

append_allow_trust_op(*trustor, asset_code, authorize, source=None*)

Append an *AllowTrust* operation to the list of operations.

Parameters

- **trustor** (*str*) – The account of the recipient of the trustline.
- **asset_code** (*str*) – The asset of the trustline the source account is authorizing. For example, if an anchor wants to allow another account to hold its USD credit, the type is USD:anchor.
- **authorize** (*Union[TrustLineEntryFlag, bool]*) – *True* to authorize the line, *False* to deauthorize if you need further control, you can also use *stellar_sdk.operation.allow_trust.TrustLineEntryFlag*.
- **source** (*Union[MixedAccount, str, None]*) – The source address that is establishing the trust in the allow trust operation.

Return type *TransactionBuilder*

Returns This builder instance.

append_begin_sponsoring_future_reserves_op(*sponsored_id, source=None*)

Append a *BeginSponsoringFutureReserves* operation to the list of operations.

Parameters

- **sponsored_id** (*str*) – The sponsored account id.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_bump_sequence_op(*bump_to, source=None*)

Append a *BumpSequence* operation to the list of operations.

Parameters

- **bump_to** (*int*) – Sequence number to bump to.
- **source** (*Union[MixedAccount, str, None]*) – The source address that is running the inflation operation.

Return type *TransactionBuilder*

Returns This builder instance.

append_change_trust_op(*asset*, *limit=None*, *source=None*)

Append a [ChangeTrust](#) operation to the list of operations.

Parameters

- **asset** ([Union\[Asset, LiquidityPoolAsset\]](#)) – The asset for the trust line.
- **limit** ([Union\[str, Decimal, None\]](#)) – The limit for the asset, defaults to `max int64(922337203685.4775807)`. If the limit is set to "0" it deletes the trustline.
- **source** ([Union\[MuxedAccount, str, None\]](#)) – The source address to add the trustline to.

Return type [TransactionBuilder](#)

Returns This builder instance.

append_claim_claimable_balance_op(*balance_id*, *source=None*)

Append a [ClaimClaimableBalance](#) operation to the list of operations.

Parameters

- **balance_id** ([str](#)) – The claimable balance id to be claimed.
- **source** ([Union\[MuxedAccount, str, None\]](#)) – The source account for the operation. Defaults to the transaction's source account.

Return type [TransactionBuilder](#)

append_clawback_claimable_balance_op(*balance_id*, *source=None*)

Append an [ClawbackClaimableBalance](#) operation to the list of operations.

Parameters

- **balance_id** ([str](#)) – The claimable balance ID to be clawed back.
- **source** ([Union\[MuxedAccount, str, None\]](#)) – The source account for the operation. Defaults to the transaction's source account.

Return type [TransactionBuilder](#)

Returns This builder instance.

append_clawback_op(*asset*, *from_*, *amount*, *source=None*)

Append an [Clawback](#) operation to the list of operations.

Parameters

- **asset** ([Asset](#)) – The asset being clawed back.
- **from_** – The public key of the account to claw back from.
- **amount** ([Union\[str, Decimal\]](#)) – The amount of the asset to claw back.
- **source** ([Union\[MuxedAccount, str, None\]](#)) – The source account for the operation. Defaults to the transaction's source account.

Return type [TransactionBuilder](#)

append_create_account_op(*destination*, *starting_balance*, *source=None*)

Append a [CreateAccount](#) operation to the list of operations.

Parameters

- **destination** ([str](#)) – Account address that is created and funded.

- **starting_balance** (`Union[str, Decimal]`) – Amount of XLM to send to the newly created account. This XLM comes from the source account.
- **source** (`Union[MixedAccount, str, None]`) – The source address to deduct funds from to fund the new account.

Return type `TransactionBuilder`

Returns This builder instance.

append_create_claimable_balance_op(*asset, amount, claimants, source=None*)

Append a `CreateClaimableBalance` operation to the list of operations.

Parameters

- **asset** (`Asset`) – The asset for the claimable balance.
- **amount** (`Union[str, Decimal]`) – the amount of the asset.
- **claimants** (`List[Claimant]`) – A list of Claimants.
- **source** (`Union[MixedAccount, str, None]`) – The source account for the operation. Defaults to the transaction's source account.

Return type `TransactionBuilder`

append_create_passive_sell_offer_op(*selling, buying, amount, price, source=None*)

Append a `CreatePassiveSellOffer` operation to the list of operations.

Parameters

- **selling** (`Asset`) – What you're selling.
- **buying** (`Asset`) – What you're buying.
- **amount** (`Union[str, Decimal]`) – The total amount you're selling.
- **price** (`Union[Price, str, Decimal]`) – Price of 1 unit of *selling* in terms of *buying*.
- **source** (`Union[MixedAccount, str, None]`) – The source account for the operation. Defaults to the transaction's source account.

Return type `TransactionBuilder`

Returns This builder instance.

append_ed25519_public_key_signer(*account_id, weight, source=None*)

Add a ed25519 public key signer to an account via a `SetOptions <stellar_sdk.operation.SetOptions>` operation. This is a helper function for `append_set_options_op()`.

Parameters

- **account_id** (`str`) – The account id of the new ed25519_public_key signer. (ex. "GDNA2V62PVEFBZ74CDJKTUHLY4Y7PL5UAV2MAM4VWF6USFE3SH2354AD")
- **weight** (`int`) – The weight of the new signer.
- **source** (`Union[MixedAccount, str, None]`) – The source account that is adding a signer to its list of signers.

Return type `TransactionBuilder`

Returns This builder instance.

append_end_sponsoring_future_reserves_op(*source=None*)

Append a `EndSponsoringFutureReserves` operation to the list of operations.

Parameters `source` (`Union[MixedAccount, str, None]`) – The source account for the operation. Defaults to the transaction’s source account.

Return type `TransactionBuilder`

Returns This builder instance.

append_hashx_signer(`sha256_hash`, `weight`, `source=None`)

Add a sha256 hash(HashX) signer to an account via a `SetOptions` <stellar_sdk.operation.SetOptions operation. This is a helper function for `append_set_options_op()`.

Parameters

- **sha256_hash** (`Union[bytes, str]`) – The address of the new sha256 hash(hashX) signer, a 32 byte hash, hex encoded string or encode strkey. (ex. "XDNA2V62PVEFBZ74CDJKTUHLY4Y7PL5UAV2MAM4VWF6USFE3SH235FXL", "da0d57da7d4850e7fc10d2a9d0ebc731f7afb40574c03395b17d49149b91f5be" or bytes)
- **weight** (`int`) – The weight of the new signer.
- **source** (`Union[MixedAccount, str, None]`) – The source account that is adding a signer to its list of signers.

Return type `TransactionBuilder`

Returns This builder instance.

append_inflation_op(`source=None`)

Append a `Inflation` operation to the list of operations.

Parameters `source` (`Union[MixedAccount, str, None]`) – The source address that is running the inflation operation.

Return type `TransactionBuilder`

Returns This builder instance.

append_liquidity_pool_deposit_op(`liquidity_pool_id`, `max_amount_a`, `max_amount_b`, `min_price`, `max_price`, `source=None`)

Append an `LiquidityPoolDeposit` operation to the list of operations.

Parameters

- **liquidity_pool_id** (`str`) – The liquidity pool ID.
- **max_amount_a** (`Union[str, Decimal]`) – Maximum amount of first asset to deposit.
- **max_amount_b** (`Union[str, Decimal]`) – Maximum amount of second asset to deposit.
- **min_price** (`Union[str, Decimal, Price]`) – Minimum deposit_a/deposit_b price.
- **max_price** (`Union[str, Decimal, Price]`) – Maximum deposit_a/deposit_b price.
- **source** (`Union[MixedAccount, str, None]`) – The source account for the operation. Defaults to the transaction’s source account.

Return type `TransactionBuilder`

Returns This builder instance.

append_liquidity_pool_withdraw_op(`liquidity_pool_id`, `amount`, `min_amount_a`, `min_amount_b`, `source=None`)

Append an `LiquidityPoolWithdraw` operation to the list of operations.

Parameters

- **liquidity_pool_id** (*str*) – The liquidity pool ID.
- **amount** (*Union[str, Decimal]*) – Amount of pool shares to withdraw.
- **min_amount_a** (*Union[str, Decimal]*) – Minimum amount of first asset to withdraw.
- **min_amount_b** (*Union[str, Decimal]*) – Minimum amount of second asset to withdraw.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction’s source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_manage_buy_offer_op(*selling, buying, amount, price, offer_id=0, source=None*)

Append a *ManageBuyOffer* operation to the list of operations.

Parameters

- **selling** (*Asset*) – What you’re selling.
- **buying** (*Asset*) – What you’re buying.
- **amount** (*Union[str, Decimal]*) – Amount being bought. if set to \emptyset , delete the offer.
- **price** (*Union[Price, str, Decimal]*) – Price of thing being bought in terms of what you are selling.
- **offer_id** (*int*) – If \emptyset , will create a new offer (default). Otherwise, edits an existing offer.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction’s source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_manage_data_op(*data_name, data_value, source=None*)

Append a *ManageData* operation to the list of operations.

Parameters

- **data_name** (*str*) – If this is a new Name it will add the given name/value pair to the account. If this Name is already present then the associated value will be modified. Up to 64 bytes long.
- **data_value** (*Union[str, bytes, None]*) – If not present then the existing *data_name* will be deleted. If present then this value will be set in the *DataEntry*. Up to 64 bytes long.
- **source** (*Union[MixedAccount, str, None]*) – The source account on which data is being managed. operation.

Return type *TransactionBuilder*

Returns This builder instance.

append_manage_sell_offer_op(*selling, buying, amount, price, offer_id=0, source=None*)

Append a *ManageSellOffer* operation to the list of operations.

Parameters

- **selling** (*Asset*) – What you’re selling.
- **buying** (*Asset*) – What you’re buying.
- **amount** (*Union[str, Decimal]*) – The total amount you’re selling. If \emptyset , deletes the offer.

- **price** (`Union[Price, str, Decimal]`) – Price of 1 unit of *selling* in terms of *buying*.
- **offer_id** (`int`) – If 0, will create a new offer (default). Otherwise, edits an existing offer.
- **source** (`Union[MuxedAccount, str, None]`) – The source account for the operation. Defaults to the transaction’s source account.

Return type `TransactionBuilder`

Returns This builder instance.

append_operation(*operation*)

Add an operation to the builder instance

Parameters **operation** (`Operation`) – an operation

Return type `TransactionBuilder`

Returns This builder instance.

append_path_payment_strict_receive_op(*destination, send_asset, send_max, dest_asset, dest_amount, path, source=None*)

Append a `PathPaymentStrictReceive` operation to the list of operations.

Parameters

- **destination** (`Union[MuxedAccount, str]`) – The destination account to send to.
- **send_asset** (`Asset`) – The *asset* to pay with.
- **send_max** (`Union[str, Decimal]`) – The maximum amount of *send_asset* to send.
- **dest_asset** (`Asset`) – The asset the *destination* will receive.
- **dest_amount** (`Union[str, Decimal]`) – The amount the *destination* receives.
- **path** (`List[Asset]`) – A list of `Asset` objects to use as the path.
- **source** (`Union[MuxedAccount, str, None]`) – The source account for the operation. Defaults to the transaction’s source account.

Return type `TransactionBuilder`

Returns This builder instance.

append_path_payment_strict_send_op(*destination, send_asset, send_amount, dest_asset, dest_min, path, source=None*)

Append a `PathPaymentStrictSend` operation to the list of operations.

Parameters

- **destination** (`Union[MuxedAccount, str]`) – The destination account to send to.
- **send_asset** (`Asset`) – The *asset* to pay with.
- **send_amount** (`Union[str, Decimal]`) – Amount of *send_asset* to send.
- **dest_asset** (`Asset`) – The asset the *destination* will receive.
- **dest_min** (`Union[str, Decimal]`) – The minimum amount of *dest_asset* to be received.
- **path** (`List[Asset]`) – A list of `Asset` objects to use as the path.
- **source** (`Union[MuxedAccount, str, None]`) – The source account for the operation. Defaults to the transaction’s source account.

Return type `TransactionBuilder`

Returns This builder instance.

append_payment_op(*destination, asset, amount, source=None*)

Append a *Payment* operation to the list of operations.

Parameters

- **destination** (*Union[MixedAccount, str]*) – The destination account ID.
- **asset** (*Asset*) – The asset to send.
- **amount** (*Union[str, Decimal]*) – The amount to send.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_pre_auth_tx_signer(*pre_auth_tx_hash, weight, source=None*)

Add a PreAuthTx signer to an account via a *SetOptions* <*stellar_sdk.operation.SetOptions* operation. This is a helper function for *append_set_options_op()*.

Parameters

- **pre_auth_tx_hash** (*Union[str, bytes]*) – The address of the new preAuthTx signer - obtained by calling *hash* on the *TransactionEnvelope*, a 32 byte hash, hex encoded string or encode strkey. (ex. "TDNA2V62PVEFBZ74CDJKTUHLY4Y7PL5UAV2MAM4VWF6USFE3SH234BSS", "da0d57da7d4850e7fc10d2a9d0ebc731f7afb40574c03395b17d49149b91f5be" or bytes)
- **weight** (*int*) – The weight of the new signer.
- **source** – The source account that is adding a signer to its list of signers.

Return type *TransactionBuilder*

Returns This builder instance.

append_revoke_account_sponsorship_op(*account_id, source=None*)

Append a *RevokeSponsorship* operation for an account to the list of operations.

Parameters

- **account_id** (*str*) – The sponsored account ID.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_revoke_claimable_balance_sponsorship_op(*claimable_balance_id, source=None*)

Append a *RevokeSponsorship* operation for a claimable to the list of operations.

Parameters

- **claimable_balance_id** (*str*) – The sponsored claimable balance ID.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_revoke_data_sponsorship_op(*account_id*, *data_name*, *source=None*)

Append a *RevokeSponsorship* operation for a data entry to the list of operations.

Parameters

- **account_id** (*str*) – The account ID which owns the data entry.
- **data_name** (*str*) – The name of the data entry
- **source** (*Union[MuxedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_revoke_ed25519_public_key_signer_sponsorship_op(*account_id*, *signer_key*, *source=None*)

Append a *RevokeSponsorship* operation for a ed25519_public_key signer to the list of operations.

Parameters

- **account_id** (*str*) – The account ID where the signer sponsorship is being removed from.
- **signer_key** (*str*) – The account id of the ed25519_public_key signer. (ex. "GDNA2V62PVEFBZ74CDJKTUHLY4Y7PL5UAV2MAM4VWF6USFE3SH2354AD")
- **source** (*Union[MuxedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_revoke_hashx_signer_sponsorship_op(*account_id*, *signer_key*, *source=None*)

Append a *RevokeSponsorship* operation for a hashx signer to the list of operations.

Parameters

- **account_id** (*str*) – The account ID where the signer sponsorship is being removed from.
- **signer_key** (*Union[bytes, str]*) – The account id of the hashx signer. (ex. "XDNA2V62PVEFBZ74CDJKTUHLY4Y7PL5UAV2MAM4VWF6USFE3SH235FXL", "da0d57da7d4850e7fc10d2a9d0ebc731f7afb40574c03395b17d49149b91f5be" or bytes)
- **source** (*Union[MuxedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_revoke_liquidity_pool_sponsorship_op(*liquidity_pool_id*, *source=None*)

Append a *RevokeSponsorship* operation for a claimable to the list of operations.

Parameters

- **liquidity_pool_id** (*str*) – The sponsored liquidity pool ID in hex string.
- **source** (*Union[MuxedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_revoke_offer_sponsorship_op(*seller_id*, *offer_id*, *source=None*)

Append a *RevokeSponsorship* operation for an offer to the list of operations.

Parameters

- **seller_id** (*str*) – The account ID which created the offer.
- **offer_id** (*int*) – The offer ID.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_revoke_pre_auth_tx_signer_sponsorship_op(*account_id*, *signer_key*, *source=None*)

Append a *RevokeSponsorship* operation for a pre_auth_tx signer to the list of operations.

Parameters

- **account_id** (*str*) – The account ID where the signer sponsorship is being removed from.
- **signer_key** (*Union[bytes, str]*) – The account id of the pre_auth_tx signer. (ex. "TDNA2V62PVEFBZ74CDJKTUHLY4Y7PL5UAV2MAM4VWF6USFE3SH234BSS", "da0d57da7d4850e7fc10d2a9d0ebc731f7afb40574c03395b17d49149b91f5be" or bytes)
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_revoke_trustline_sponsorship_op(*account_id*, *asset*, *source=None*)

Append a *RevokeSponsorship* operation for a trustline to the list of operations.

Parameters

- **account_id** (*str*) – The account ID which owns the trustline.
- **asset** (*Union[Asset, LiquidityPoolId]*) – The asset in the trustline.
- **source** (*Union[MixedAccount, str, None]*) – The source account for the operation. Defaults to the transaction's source account.

Return type *TransactionBuilder*

Returns This builder instance.

append_set_options_op(*inflation_dest=None*, *clear_flags=None*, *set_flags=None*, *master_weight=None*, *low_threshold=None*, *med_threshold=None*, *high_threshold=None*, *home_domain=None*, *signer=None*, *source=None*)

Append a *SetOptions* operation to the list of operations.

Parameters

- **inflation_dest** (*Optional[str]*) – Account of the inflation destination.
- **clear_flags** (*Union[int, AuthorizationFlag, None]*) – Indicates which flags to clear. For details about the flags, please refer to the [Control Access to an Asset - Flag](#). The bit mask integer subtracts from the existing flags of the account. This allows for setting specific bits without knowledge of existing flags, you can also use *stellar_sdk.operation.set_options.AuthorizationFlag*

- AUTHORIZATION_REQUIRED = 1
- AUTHORIZATION_REVOCABLE = 2
- AUTHORIZATION_IMMUTABLE = 4
- AUTHORIZATION_CLAWBACK_ENABLED = 8
- **set_flags** (`Union[int, AuthorizationFlag, None]`) – Indicates which flags to set. For details about the flags, please refer to the [Control Access to an Asset - Flag](#). The bit mask integer adds onto the existing flags of the account. This allows for setting specific bits without knowledge of existing flags, you can also use `stellar_sdk.operation.set_options.AuthorizationFlag`
 - AUTHORIZATION_REQUIRED = 1
 - AUTHORIZATION_REVOCABLE = 2
 - AUTHORIZATION_IMMUTABLE = 4
 - AUTHORIZATION_CLAWBACK_ENABLED = 8
- **master_weight** (`Optional[int]`) – A number from 0-255 (inclusive) representing the weight of the master key. If the weight of the master key is updated to 0, it is effectively disabled.
- **low_threshold** (`Optional[int]`) – A number from 0-255 (inclusive) representing the threshold this account sets on all operations it performs that have a [low threshold](#).
- **med_threshold** (`Optional[int]`) – A number from 0-255 (inclusive) representing the threshold this account sets on all operations it performs that have a [medium threshold](#).
- **high_threshold** (`Optional[int]`) – A number from 0-255 (inclusive) representing the threshold this account sets on all operations it performs that have a [high threshold](#).
- **home_domain** (`Optional[str]`) – sets the home domain used for reverse [federation](#) lookup.
- **signer** (`Optional[Signer]`) – Add, update, or remove a signer from the account.
- **source** (`Union[MuxedAccount, str, None]`) – The source account for the operation. Defaults to the transaction’s source account.

Return type `TransactionBuilder`

Returns This builder instance.

append_set_trust_line_flags_op(`trustor, asset, clear_flags=None, set_flags=None, source=None`)
Append an `SetTrustLineFlags` operation to the list of operations.

Parameters

- **trustor** (`str`) – The account whose trustline this is.
- **asset** (`Asset`) – The asset on the trustline.
- **clear_flags** (`Optional[TrustLineFlags]`) – The flags to clear.
- **set_flags** (`Optional[TrustLineFlags]`) – The flags to set.
- **source** (`Union[MuxedAccount, str, None]`) – The source account for the operation. Defaults to the transaction’s source account.

Return type `TransactionBuilder`

Returns This builder instance.

build()

This will build the transaction envelope. It will also increment the source account's sequence number by 1.

Return type *TransactionEnvelope*

Returns New transaction envelope.

static build_fee_bump_transaction(*fee_source*, *base_fee*, *inner_transaction_envelope*,
network_passphrase='Test SDF Network ; September 2015')

Create a *FeeBumpTransactionEnvelope* object.

See [CAP-0015](#) for more information.

Parameters

- **fee_source** (*Union*[*MuxedAccount*, *Keypair*, *str*]) – The account paying for the transaction.
- **base_fee** (*int*) – The max fee willing to pay per operation in inner transaction (**in stroops**).
- **inner_transaction_envelope** (*TransactionEnvelope*) – The *TransactionEnvelope* to be bumped by the fee bump transaction.
- **network_passphrase** (*str*) – The network to connect to for verifying and retrieving additional attributes from.

Return type *FeeBumpTransactionEnvelope*

Returns a *TransactionBuilder* via the XDR object.

static from_xdr(*xdr*, *network_passphrase*)

Create a *TransactionBuilder* or *FeeBumpTransactionEnvelope* via an XDR object.

Warning: I don't recommend you to use this function, because it loses its signature when constructing *TransactionBuilder*. Please use *stellar_sdk.helpers.parse_transaction_envelope_from_xdr()* instead.

In addition, if *xdr* is not of *TransactionEnvelope*, it sets the fields of this builder (the transaction envelope, transaction, operations, source, etc.) to all of the fields in the provided XDR transaction envelope, but the signature will not be added to it.

Parameters

- **xdr** (*str*) – The XDR object representing the transaction envelope to which this builder is setting its state to.
- **network_passphrase** (*str*) – The network to connect to for verifying and retrieving additional attributes from.

Return type *Union*[*TransactionBuilder*, *FeeBumpTransactionEnvelope*]

Returns a *TransactionBuilder* or *FeeBumpTransactionEnvelope* via the XDR object.

set_timeout(*timeout*)

Set timeout for the transaction, actually set a *TimeBounds*.

Parameters **timeout** (*int*) – timeout in second.

Return type *TransactionBuilder*

Returns This builder instance.

Raises *ValueError*: if *time_bound* is already set.

2.1.27 Helpers

`stellar_sdk.helpers.parse_transaction_envelope_from_xdr(xdr, network_passphrase)`

When you are not sure whether your XDR belongs to `TransactionEnvelope` or `FeeBumpTransactionEnvelope`, you can use this helper function.

An example:

```
from stellar_sdk import Network
from stellar_sdk.helpers import parse_transaction_envelope_from_xdr

xdr = "AAAAAgAAAADHJNEDn33/C1uDkDfzDfKvq/
↳4XE9IxGfGiLCfoV7riZQAAA+gCI4TVABpRPgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADAAAAUxpcmEAAAAAabIaDgm0ypyJpsVfEj
↳kEB2Z4UL20y536evnwmmSc4c2FnxlVUcPZl5jgWHcNwY8LTpFhdrUN9TZWciCRp/JCZYa0SJh8cYB"
te = parse_transaction_envelope_from_xdr(xdr, Network.PUBLIC_NETWORK_PASSPHRASE)
print(te)
```

Parameters

- **xdr** (`str`) – Transaction envelope XDR
- **network_passphrase** (`str`) – The network to connect to for verifying and retrieving additional attributes from. (ex. "Public Global Stellar Network ; September 2015")

Raises `ValueError` – XDR is neither `TransactionEnvelope` nor `FeeBumpTransactionEnvelope`

Return type `Union[TransactionEnvelope, FeeBumpTransactionEnvelope]`

2.1.28 XDR Utils

`stellar_sdk.xdr.utils.from_xdr_amount(value)`

Converts an str amount from an XDR amount object

Parameters **value** (`int`) – The amount to convert to a string from an XDR int64 amount.

Return type `str`

`stellar_sdk.xdr.utils.to_xdr_amount(value)`

Converts an amount to the appropriate value to send over the network as a part of an XDR object.

Each asset amount is encoded as a signed 64-bit integer in the XDR structures. An asset amount unit (that which is seen by end users) is scaled down by a factor of ten million (10,000,000) to arrive at the native 64-bit integer representation. For example, the integer amount value 25,123,456 equals 2.5123456 units of the asset. This scaling allows for seven decimal places of precision in human-friendly amount units.

This static method correctly multiplies the value by the scaling factor in order to come to the integer value used in XDR structures.

See [Stellar's documentation on Asset Precision](#) for more information.

Parameters **value** (`Union[str, Decimal]`) – The amount to convert to an integer for XDR serialization.

Return type `int`

2.1.29 Stellar Ecosystem Proposals

SEP 0001: stellar.toml

`stellar_sdk.sep.stellar_toml.fetch_stellar_toml(domain, client=None, use_http=False)`

Retrieve the stellar.toml file from a given domain.

Retrieve the stellar.toml file for information about interacting with Stellar's federation protocol for a given Stellar Anchor (specified by a domain).

Parameters

- **domain** (`str`) – The domain the .toml file is hosted at.
- **use_http** (`bool`) – Specifies whether the request should go over plain HTTP vs HTTPS. Note it is recommend that you **always** use HTTPS.
- **client** (`Optional[BaseSyncClient]`) – Http Client used to send the request.

Return type `MutableMapping[str, Any]`

Returns The stellar.toml file as an object via `toml.loads()`.

Raises `StellarTomlNotFoundError`: if the Stellar toml file could not not be found.

`async stellar_sdk.sep.stellar_toml.fetch_stellar_toml_async(domain, client=None, use_http=False)`

Retrieve the stellar.toml file from a given domain.

Retrieve the stellar.toml file for information about interacting with Stellar's federation protocol for a given Stellar Anchor (specified by a domain).

Parameters

- **domain** (`str`) – The domain the .toml file is hosted at.
- **use_http** (`bool`) – Specifies whether the request should go over plain HTTP vs HTTPS. Note it is recommend that you **always** use HTTPS.
- **client** (`Optional[BaseAsyncClient]`) – Http Client used to send the request.

Return type `MutableMapping[str, Any]`

Returns The stellar.toml file as an object via `toml.loads()`.

Raises `StellarTomlNotFoundError`: if the Stellar toml file could not not be found.

SEP 0002: Federation protocol

`stellar_sdk.sep.federation.resolve_stellar_address(stellar_address, client=None, federation_url=None, use_http=False)`

Get the federation record if the user was found for a given Stellar address.

Parameters

- **stellar_address** (`str`) – address Stellar address (ex. "bob*stellar.org").
- **client** (`Optional[BaseSyncClient]`) – Http Client used to send the request.
- **federation_url** (`Optional[str]`) – The federation server URL (ex. "https://stellar.org/federation"), if you don't set this value, we will try to get it from `stellar_address`.

- **use_http** (*bool*) – Specifies whether the request should go over plain HTTP vs HTTPS. Note it is recommend that you **always** use HTTPS.

Return type *FederationRecord*

Returns Federation record.

```
async stellar_sdk.sep.federation.resolve_stellar_address_async(stellar_address, client=None,
                                                             federation_url=None,
                                                             use_http=False)
```

Get the federation record if the user was found for a given Stellar address.

Parameters

- **stellar_address** (*str*) – address Stellar address (ex. "bob*stellar.org").
- **client** (*Optional[BaseAsyncClient]*) – Http Client used to send the request.
- **federation_url** (*Optional[str]*) – The federation server URL (ex. "https://stellar.org/federation"), if you don't set this value, we will try to get it from *stellar_address*.
- **use_http** (*bool*) – Specifies whether the request should go over plain HTTP vs HTTPS. Note it is recommend that you **always** use HTTPS.

Return type *FederationRecord*

Returns Federation record.

```
async stellar_sdk.sep.federation.resolve_account_id_async(account_id, domain=None,
                                                         federation_url=None, client=None,
                                                         use_http=False)
```

Given an account ID, get their federation record if the user was found

Parameters

- **account_id** (*str*) – Account ID (ex. "GBYNR2QJXLBCBTRN44MRORCMI4YO7FZPFBCNOKTOBCAAFC7KC3LNPRYS")
- **domain** (*Optional[str]*) – Get *federation_url* from the domain, you don't need to set this value if *federation_url* is set.
- **federation_url** (*Optional[str]*) – The federation server URL (ex. "https://stellar.org/federation").
- **client** (*Optional[BaseAsyncClient]*) – Http Client used to send the request.
- **use_http** (*bool*) – Specifies whether the request should go over plain HTTP vs HTTPS. Note it is recommend that you **always** use HTTPS.

Return type *FederationRecord*

Returns Federation record.

```
stellar_sdk.sep.federation.resolve_account_id(account_id, domain=None, federation_url=None,
                                              client=None, use_http=False)
```

Given an account ID, get their federation record if the user was found

Parameters

- **account_id** (*str*) – Account ID (ex. "GBYNR2QJXLBCBTRN44MRORCMI4YO7FZPFBCNOKTOBCAAFC7KC3LNPRYS")
- **domain** (*Optional[str]*) – Get *federation_url* from the domain, you don't need to set this value if *federation_url* is set.
- **federation_url** (*Optional[str]*) – The federation server URL (ex. "https://stellar.org/federation").

- **client** (*Optional*[*BaseSyncClient*]) – Http Client used to send the request.
- **use_http** (*bool*) – Specifies whether the request should go over plain HTTP vs HTTPS. Note it is recommend that you **always** use HTTPS.

Return type *FederationRecord*

Returns Federation record.

```
class stellar_sdk.sep.federation.FederationRecord(account_id, stellar_address, memo_type, memo)
```

SEP 0005: Key Derivation Methods for Stellar Accounts

```
class stellar_sdk.sep.mnemonic.StellarMnemonic(language=Language.ENGLISH)
    Please use stellar_sdk.keypair.Keypair.generate_mnemonic_phrase() and stellar_sdk.keypair.Keypair.from_mnemonic_phrase()
```

```
class stellar_sdk.sep.mnemonic.Language(value)
    The type of language supported by the mnemonic.
```

```
CHINESE_SIMPLIFIED = 'chinese_simplified'
```

```
CHINESE_TRADITIONAL = 'chinese_traditional'
```

```
ENGLISH = 'english'
```

```
FRENCH = 'french'
```

```
ITALIAN = 'italian'
```

```
JAPANESE = 'japanese'
```

```
KOREAN = 'korean'
```

```
SPANISH = 'spanish'
```

SEP 0007: URI Scheme to facilitate delegated signing

```
class stellar_sdk.sep.stellar_uri.PayStellarUri(destination, amount=None, asset=None,
                                                memo=None, callback=None, message=None,
                                                network_passphrase=None, origin_domain=None,
                                                signature=None)
```

A request for a payment to be signed.

See SEP-0007

Parameters

- **destination** (*str*) – A valid account ID or payment address.
- **amount** (*Optional*[*str*]) – Amount that destination will receive.
- **asset** (*Optional*[*Asset*]) – Asset destination will receive.
- **memo** (*Optional*[*Memo*]) – A memo to attach to the transaction.
- **callback** (*Optional*[*str*]) – The uri to post the transaction to after signing.
- **message** (*Optional*[*str*]) – An message for displaying to the user.
- **network_passphrase** (*Optional*[*str*]) – The passphrase of the target network.
- **origin_domain** (*Optional*[*str*]) – A fully qualified domain name that specifies the originating domain of the URI request.

- **signature** (`Optional[str]`) – A base64 encode signature of the hash of the URI request.

classmethod from_uri (*cls, uri*)

Parse Stellar Pay URI and generate *PayStellarUri* object.

Parameters **uri** (`str`) – Stellar Pay URI.

Return type *PayStellarUri*

Returns *PayStellarUri* object from uri.

sign (*signer*)

Sign the URI.

Parameters **signer** (`Union[Keypair, str]`) – The account used to sign this transaction, it should be the secret key of *URI_REQUEST_SIGNING_KEY*.

Return type `None`

to_uri ()

Generate the request URI.

Return type `str`

Returns Stellar Pay URI.

```
class stellar_sdk.sep.stellar_uri.TransactionStellarUri (transaction_envelope, replace=None,
                                                         callback=None, pubkey=None,
                                                         message=None,
                                                         network_passphrase=None,
                                                         origin_domain=None, signature=None)
```

A request for a transaction to be signed.

See [SEP-0007](#)

Parameters

- **transaction_envelope** (`Union[TransactionEnvelope, FeeBumpTransactionEnvelope]`) – Transaction waiting to be signed.
- **replace** (`Optional[List[Replacement]]`) – A value that identifies the fields to be replaced in the xdr using the Txrep (SEP-0011) representation.
- **callback** (`Optional[str]`) – The uri to post the transaction to after signing.
- **pubkey** (`Optional[str]`) – Specify which public key you want the URI handler to sign for.
- **message** (`Optional[str]`) – An message for displaying to the user.
- **network_passphrase** (`Optional[str]`) – The passphrase of the target network.
- **origin_domain** (`Optional[str]`) – A fully qualified domain name that specifies the originating domain of the URI request.
- **signature** (`Optional[str]`) – A base64 encode signature of the hash of the URI request.

classmethod from_uri (*cls, uri, network_passphrase*)

Parse Stellar Transaction URI and generate *TransactionStellarUri* object.

Parameters

- **uri** (`str`) – Stellar Transaction URI.
- **network_passphrase** (`Optional[str]`) – The network to connect to for verifying and retrieving xdr, If it is set to *None*, the *network_passphrase* in the uri will not be verified.

Return type *TransactionStellarUri*

Returns `TransactionStellarUri` object from uri.

sign(*signer*)

Sign the URI.

Parameters **signer** (`Union[Keypair, str]`) – The account used to sign this transaction, it should be the secret key of `URI_REQUEST_SIGNING_KEY`.

Return type `None`

to_uri()

Generate the request URI.

Return type `str`

Returns Stellar Transaction URI.

class `stellar_sdk.sep.stellar_uri.Replacement`(*txrep_tx_field_name*, *reference_identifier*, *hint*)

Used to represent a single replacement.

An example:

```
r1 = Replacement("sourceAccount", "X", "account on which to create the trustline")
r2 = Replacement("seqNum", "Y", "sequence for sourceAccount")
replacements = [r1, r2]
```

See [SEP-0007](#)

Parameters

- **txrep_tx_field_name** (`str`) – Txrep tx field name.
- **reference_identifier** (`str`) – Reference identifier.
- **hint** (`str`) – A brief and clear explanation of the context for the *reference_identifier*.

SEP 0010: Stellar Web Authentication

`stellar_sdk.sep.stellar_web_authentication.build_challenge_transaction`(*server_secret*,
client_account_id,
home_domain,
web_auth_domain,
network_passphrase,
timeout=900,
client_domain=None,
client_signing_key=None,
memo=None)

Returns a valid [SEP0010](#) challenge transaction which you can use for Stellar Web Authentication.

Parameters

- **server_secret** (`str`) – secret key for server's stellar.toml `SIGNING_KEY`.
- **client_account_id** (`str`) – The stellar account (G. . .) or muxed account (M. . .) that the wallet wishes to authenticate with the server.
- **home_domain** (`str`) – The fully qualified domain name of the service requiring authentication (ex. "example.com").
- **web_auth_domain** (`str`) – The fully qualified domain name of the service issuing the challenge.

- **network_passphrase** (`str`) – The network to connect to for verifying and retrieving additional attributes from. (ex. "Public Global Stellar Network ; September 2015")
- **timeout** (`int`) – Challenge duration in seconds (default to 15 minutes).
- **client_domain** (`Optional[str]`) – The domain of the client application requesting authentication
- **client_signing_key** (`Optional[str]`) – The stellar account listed as the SIGNING_KEY on the client domain's TOML file
- **memo** (`Optional[int]`) – The ID memo to attach to the transaction. Not permitted if `client_account_id` is a muxed account

Return type `str`

Returns A base64 encoded string of the raw TransactionEnvelope xdr struct for the transaction.

```
stellar_sdk.sep.stellar_web_authentication.read_challenge_transaction(challenge_transaction,
                                                                    server_account_id,
                                                                    home_domains,
                                                                    web_auth_domain,
                                                                    network_passphrase)
```

Reads a SEP 10 challenge transaction and returns the decoded transaction envelope and client account ID contained within.

It also verifies that transaction is signed by the server.

It does not verify that the transaction has been signed by the client or that any signatures other than the servers on the transaction are valid. Use one of the following functions to completely verify the transaction:

- `stellar_sdk.sep.stellar_web_authentication.verify_challenge_transaction_threshold()`
- `stellar_sdk.sep.stellar_web_authentication.verify_challenge_transaction_signers()`

Parameters

- **challenge_transaction** (`str`) – SEP0010 transaction challenge transaction in base64.
- **server_account_id** (`str`) – public key for server's account.
- **home_domains** (`Union[str, Iterable[str]]`) – The home domain that is expected to be included in the first Manage Data operation's string key. If a list is provided, one of the domain names in the array must match.
- **web_auth_domain** (`str`) – The home domain that is expected to be included as the value of the Manage Data operation with the 'web_auth_domain' key. If no such operation is included, this parameter is not used.
- **network_passphrase** (`str`) – The network to connect to for verifying and retrieving additional attributes from. (ex. "Public Global Stellar Network ; September 2015")

Raises `InvalidSep10ChallengeError` - if the validation fails, the exception will be thrown.

Return type `ChallengeTransaction`

`stellar_sdk.sep.stellar_web_authentication.verify_challenge_transaction_threshold`(*challenge_transaction*, *server_account_id*, *home_domains*, *web_auth_domain*, *network_passphrase*, *threshold*, *signers*)

Verifies that for a SEP 10 challenge transaction all signatures on the transaction are accounted for and that the signatures meet a threshold on an account. A transaction is verified if it is signed by the server account, and all other signatures match a signer that has been provided as an argument, and those signatures meet a threshold on the account.

Parameters

- **challenge_transaction** (*str*) – SEP0010 transaction challenge transaction in base64.
- **server_account_id** (*str*) – public key for server’s account.
- **home_domains** (*Union[str, Iterable[str]]*) – The home domain that is expected to be included in the first Manage Data operation’s string key. If a list is provided, one of the domain names in the array must match.
- **web_auth_domain** (*str*) – The home domain that is expected to be included as the value of the Manage Data operation with the ‘web_auth_domain’ key. If no such operation is included, this parameter is not used.
- **network_passphrase** (*str*) – The network to connect to for verifying and retrieving additional attributes from. (ex. "Public Global Stellar Network ; September 2015")
- **threshold** (*int*) – The medThreshold on the client account.
- **signers** (*List[Ed25519PublicKeySigner]*) – The signers of client account.

Raises *InvalidSep10ChallengeError*: - The transaction is invalid according to `stellar_sdk.sep.stellar_web_authentication.read_challenge_transaction()`. - One or more signatures in the transaction are not identifiable as the server account or one of the signers provided in the arguments. - The signatures are all valid but do not meet the threshold.

Return type *List[Ed25519PublicKeySigner]*

`stellar_sdk.sep.stellar_web_authentication.verify_challenge_transaction_signed_by_client_master_key`(*challenge_transaction*, *server_account_id*, *home_domains*, *web_auth_domain*, *network_passphrase*, *signers*)

An alias for `stellar_sdk.sep.stellar_web_authentication.verify_challenge_transaction()`.

Parameters

- **challenge_transaction** (*str*) – SEP0010 transaction challenge transaction in base64.
- **server_account_id** (*str*) – public key for server’s account.
- **home_domains** (*Union[str, Iterable[str]]*) – The home domain that is expected to be included in the first Manage Data operation’s string key. If a list is provided, one of the domain names in the array must match.

- **web_auth_domain** (*str*) – The home domain that is expected to be included as the value of the Manage Data operation with the ‘web_auth_domain’ key. If no such operation is included, this parameter is not used.
- **network_passphrase** (*str*) – The network to connect to for verifying and retrieving additional attributes from. (ex. "Public Global Stellar Network ; September 2015")

Raises *InvalidSep10ChallengeError* - if the validation fails, the exception will be thrown.

Return type *None*

`stellar_sdk.sep.stellar_web_authentication.verify_challenge_transaction_signers`(*challenge_transaction*,
server_account_id,
home_domains,
web_auth_domain,
network_passphrase,
signers)

Verifies that for a SEP 10 challenge transaction all signatures on the transaction are accounted for. A transaction is verified if it is signed by the server account, and all other signatures match a signer that has been provided as an argument. Additional signers can be provided that do not have a signature, but all signatures must be matched to a signer for verification to succeed. If verification succeeds a list of signers that were found is returned, excluding the server account ID.

Parameters

- **challenge_transaction** (*str*) – SEP0010 transaction challenge transaction in base64.
- **server_account_id** (*str*) – public key for server’s account.
- **home_domains** (*Union[str, Iterable[str]]*) – The home domain that is expected to be included in the first Manage Data operation’s string key. If a list is provided, one of the domain names in the array must match.
- **web_auth_domain** (*str*) – The home domain that is expected to be included as the value of the Manage Data operation with the ‘web_auth_domain’ key, if present.
- **network_passphrase** (*str*) – The network to connect to for verifying and retrieving additional attributes from. (ex. "Public Global Stellar Network ; September 2015")
- **signers** (*List[Ed25519PublicKeySigner]*) – The signers of client account.

Raises *InvalidSep10ChallengeError*: - The transaction is invalid according to `stellar_sdk.sep.stellar_web_authentication.read_challenge_transaction()`. - One or more signatures in the transaction are not identifiable as the server account or one of the signers provided in the arguments.

Return type *List[Ed25519PublicKeySigner]*

`stellar_sdk.sep.stellar_web_authentication.verify_challenge_transaction`(*challenge_transaction*,
server_account_id,
home_domains,
web_auth_domain,
network_passphrase)

Verifies if a transaction is a valid SEP0010 v1.2 challenge transaction, if the validation fails, an exception will be thrown.

This function performs the following checks:

1. verify that transaction sequenceNumber is equal to zero;
2. verify that transaction source account is equal to the server’s signing key;

3. verify that transaction has time bounds set, and that current time is between the minimum and maximum bounds;
4. verify that transaction contains a single Manage Data operation and it's source account is not null;
5. verify that transaction envelope has a correct signature by server's signing key;
6. verify that transaction envelope has a correct signature by the operation's source account;

Parameters

- **challenge_transaction** (*str*) – SEP0010 transaction challenge transaction in base64.
- **server_account_id** (*str*) – public key for server's account.
- **home_domains** (*Union[str, Iterable[str]]*) – The home domain that is expected to be included in the first Manage Data operation's string key. If a list is provided, one of the domain names in the array must match.
- **web_auth_domain** (*str*) – The home domain that is expected to be included as the value of the Manage Data operation with the *web_auth_domain* key, if present.
- **network_passphrase** (*str*) – The network to connect to for verifying and retrieving additional attributes from. (ex. "Public Global Stellar Network ; September 2015")

Raises *InvalidSep10ChallengeError* - if the validation fails, the exception will be thrown.

Return type *None*

```
class stellar_sdk.sep.stellar_web_authentication.ChallengeTransaction(transaction,
                                                                    client_account_id,
                                                                    matched_home_domain,
                                                                    memo=None)
```

Used to store the results produced by *stellar_sdk.sep.stellar_web_authentication.read_challenge_transaction()*.

Parameters

- **transaction** (*TransactionEnvelope*) – The TransactionEnvelope parsed from challenge xdr.
- **client_account_id** (*str*) – The stellar account that the wallet wishes to authenticate with the server.
- **matched_home_domain** (*str*) – The domain name that has been matched.
- **memo** (*Optional[int]*) – The ID memo attached to the transaction

SEP 0011: Txrep: human-readable low-level representation of Stellar transactions

```
stellar_sdk.sep.txrep.to_txrep(transaction_envelope)
```

Generate a human-readable format for Stellar transactions.

MuxAccount is currently not supported.

Txrep is a human-readable representation of Stellar transactions that functions like an assembly language for XDR.

See SEP-0011

Parameters *transaction_envelope* (*Union[TransactionEnvelope, FeeBumpTransactionEnvelope]*) – Transaction envelope object.

Return type `str`

Returns A human-readable format for Stellar transactions.

`stellar_sdk.sep.txrep.from_txrep(txrep, network_passphrase)`
Parse txrep and generate transaction envelope object.

MuxAccount is currently not supported.

Txrep is a human-readable representation of Stellar transactions that functions like an assembly language for XDR.

See [SEP-0011](#)

Parameters

- **txrep** (`str`) – a human-readable format for Stellar transactions.
- **network_passphrase** (`str`) – The network to connect, you do not need to set this value at this time, it is reserved for future use.

Return type `Union[TransactionEnvelope, FeeBumpTransactionEnvelope]`

Returns A human-readable format for Stellar transactions.

Exceptions

class `stellar_sdk.sep.exceptions.StellarTomlNotFoundError`

If the SEP 0010 toml file not found, the exception will be thrown.

class `stellar_sdk.sep.exceptions.InvalidFederationAddress`

If the federation address is invalid, the exception will be thrown.

class `stellar_sdk.sep.exceptions.FederationServerNotFoundError`

If the federation address is invalid, the exception will be thrown.

class `stellar_sdk.sep.exceptions.BadFederationResponseError(response)`

If the federation address is invalid, the exception will be thrown.

Parameters `response` – client response

class `stellar_sdk.sep.exceptions.InvalidSep10ChallengeError`

If the SEP 0010 validation fails, the exception will be thrown.

class `stellar_sdk.sep.exceptions.AccountRequiresMemoError(message, account_id, operation_index)`

`AccountRequiresMemoError` is raised when a transaction is trying to submit an operation to an account which requires a memo.

This error contains two attributes to help you identify the account requiring the memo and the operation where the account is the destination.

See [SEP-0029](#) for more information.

2.1.30 stellar_sdk.xdr

AccountEntry

```
class stellar_sdk.xdr.account_entry.AccountEntry(account_id, balance, seq_num, num_sub_entries,
                                                inflation_dest, flags, home_domain, thresholds,
                                                signers, ext)
```

XDR Source Code:

```
struct AccountEntry
{
    AccountID accountID;    // master public key for this account
    int64 balance;         // in stroops
    SequenceNumber seqNum; // last sequence number used for this account
    uint32 numSubEntries;  // number of sub-entries this account has
                        // drives the reserve
    AccountID* inflationDest; // Account to vote for during inflation
    uint32 flags;          // see AccountFlags

    string32 homeDomain; // can be used for reverse federation and memo lookup

    // fields used for signatures
    // thresholds stores unsigned bytes: [weight of master|low|medium|high]
    Thresholds thresholds;

    Signer signers<MAX_SIGNERS>; // possible signers for this account

    // reserved for future use
    union switch (int v)
    {
        case 0:
            void;
        case 1:
            AccountEntryExtensionV1 v1;
    }
    ext;
};
```

AccountEntryExt

```
class stellar_sdk.xdr.account_entry_ext.AccountEntryExt(v, v1=None)
```

XDR Source Code:

```
union switch (int v)
{
    case 0:
        void;
    case 1:
        AccountEntryExtensionV1 v1;
}
```

AccountEntryExtensionV1

class stellar_sdk.xdr.account_entry_extension_v1.**AccountEntryExtensionV1**(*liabilities, ext*)
 XDR Source Code:

```
struct AccountEntryExtensionV1
{
    Liabilities liabilities;

    union switch (int v)
    {
        case 0:
            void;
        case 2:
            AccountEntryExtensionV2 v2;
    }
    ext;
};
```

AccountEntryExtensionV1Ext

class stellar_sdk.xdr.account_entry_extension_v1_ext.**AccountEntryExtensionV1Ext**(*v, v2=None*)

XDR Source Code:

```
union switch (int v)
{
    case 0:
        void;
    case 2:
        AccountEntryExtensionV2 v2;
}
```

AccountEntryExtensionV2

class stellar_sdk.xdr.account_entry_extension_v2.**AccountEntryExtensionV2**(*num_sponsored, num_sponsoring, signer_sponsoring_ids, ext*)

XDR Source Code:

```
struct AccountEntryExtensionV2
{
    uint32 numSponsored;
    uint32 numSponsoring;
    SponsorshipDescriptor signerSponsoringIDs<MAX_SIGNERS>;

    union switch (int v)
    {
        case 0:
            void;
```

(continues on next page)

(continued from previous page)

```

    }
    ext;
};

```

AccountEntryExtensionV2Ext

`class stellar_sdk.xdr.account_entry_extension_v2_ext.AccountEntryExtensionV2Ext(v)`

XDR Source Code:

```

union switch (int v)
{
    case 0:
        void;
}

```

AccountFlags

`class stellar_sdk.xdr.account_flags.AccountFlags(value)`

XDR Source Code:

```

enum AccountFlags
{ // masks for each flag

    // Flags set on issuer accounts
    // TrustLines are created with authorized set to "false" requiring
    // the issuer to set it for each TrustLine
    AUTH_REQUIRED_FLAG = 0x1,
    // If set, the authorized flag in TrustLines can be cleared
    // otherwise, authorization cannot be revoked
    AUTH_REVOCABLE_FLAG = 0x2,
    // Once set, causes all AUTH_* flags to be read-only
    AUTH_IMMUTABLE_FLAG = 0x4,
    // Trustlines are created with clawback enabled set to "true",
    // and claimable balances created from those trustlines are created
    // with clawback enabled set to "true"
    AUTH_CLAWBACK_ENABLED_FLAG = 0x8
};

```

AccountID

`class stellar_sdk.xdr.account_id.AccountID(account_id)`

XDR Source Code:

```

typedef PublicKey AccountID;

```

AccountMergeResult

```
class stellar_sdk.xdr.account_merge_result.AccountMergeResult(code,
                                                             source_account_balance=None)
```

XDR Source Code:

```
union AccountMergeResult switch (AccountMergeResultCode code)
{
  case ACCOUNT_MERGE_SUCCESS:
    int64 sourceAccountBalance; // how much got transferred from source account
  default:
    void;
};
```

AccountMergeResultCode

```
class stellar_sdk.xdr.account_merge_result_code.AccountMergeResultCode(value)
```

XDR Source Code:

```
enum AccountMergeResultCode
{
  // codes considered as "success" for the operation
  ACCOUNT_MERGE_SUCCESS = 0,
  // codes considered as "failure" for the operation
  ACCOUNT_MERGE_MALFORMED = -1, // can't merge onto itself
  ACCOUNT_MERGE_NO_ACCOUNT = -2, // destination does not exist
  ACCOUNT_MERGE_IMMUTABLE_SET = -3, // source account has AUTH_IMMUTABLE set
  ACCOUNT_MERGE_HAS_SUB_ENTRIES = -4, // account has trust lines/offers
  ACCOUNT_MERGE_SEQNUM_TOO_FAR = -5, // sequence number is over max allowed
  ACCOUNT_MERGE_DEST_FULL = -6, // can't add source balance to
  // destination balance
  ACCOUNT_MERGE_IS_SPONSOR = -7 // can't merge account that is a sponsor
};
```

AllowTrustOp

```
class stellar_sdk.xdr.allow_trust_op.AllowTrustOp(trustor, asset, authorize)
```

XDR Source Code:

```
struct AllowTrustOp
{
  AccountID trustor;
  AssetCode asset;

  // One of 0, AUTHORIZED_FLAG, or AUTHORIZED_TO_MAINTAIN_LIABILITIES_FLAG
  uint32 authorize;
};
```

AllowTrustResult

class stellar_sdk.xdr.allow_trust_result.**AllowTrustResult**(code)

XDR Source Code:

```
union AllowTrustResult switch (AllowTrustResultCode code)
{
  case ALLOW_TRUST_SUCCESS:
    void;
  default:
    void;
};
```

AllowTrustResultCode

class stellar_sdk.xdr.allow_trust_result_code.**AllowTrustResultCode**(value)

XDR Source Code:

```
enum AllowTrustResultCode
{
  // codes considered as "success" for the operation
  ALLOW_TRUST_SUCCESS = 0,
  // codes considered as "failure" for the operation
  ALLOW_TRUST_MALFORMED = -1, // asset is not ASSET_TYPE_ALPHANUM
  ALLOW_TRUST_NO_TRUST_LINE = -2, // trustor does not have a trustline
  // source account does not require trust
  ALLOW_TRUST_TRUST_NOT_REQUIRED = -3,
  ALLOW_TRUST_CANT_REVOKE = -4, // source account can't revoke trust,
  ALLOW_TRUST_SELF_NOT_ALLOWED = -5 // trusting self is not allowed
};
```

AlphaNum12

class stellar_sdk.xdr.alpha_num12.**AlphaNum12**(asset_code, issuer)

XDR Source Code:

```
struct AlphaNum12
{
  AssetCode12 assetCode;
  AccountID issuer;
};
```


AlphaNum4

class stellar_sdk.xdr.alpha_num4.**AlphaNum4**(*asset_code*, *issuer*)
XDR Source Code:

```
struct AlphaNum4
{
    AssetCode4 assetCode;
    AccountID issuer;
};
```

Asset

class stellar_sdk.xdr.asset.**Asset**(*type*, *alpha_num4=None*, *alpha_num12=None*)
XDR Source Code:

```
union Asset switch (AssetType type)
{
    case ASSET_TYPE_NATIVE: // Not credit
        void;

    case ASSET_TYPE_CREDIT_ALPHANUM4:
        AlphaNum4 alphaNum4;

    case ASSET_TYPE_CREDIT_ALPHANUM12:
        AlphaNum12 alphaNum12;

    // add other asset types here in the future
};
```

AssetCode

class stellar_sdk.xdr.asset_code.**AssetCode**(*type*, *asset_code4=None*, *asset_code12=None*)
XDR Source Code:

```
union AssetCode switch (AssetType type)
{
    case ASSET_TYPE_CREDIT_ALPHANUM4:
        AssetCode4 assetCode4;

    case ASSET_TYPE_CREDIT_ALPHANUM12:
        AssetCode12 assetCode12;

    // add other asset types here in the future
};
```

AssetCode12

class stellar_sdk.xdr.asset_code12.**AssetCode12**(*asset_code12*)
XDR Source Code:

```
typedef opaque AssetCode12[12];
```

AssetCode4

class stellar_sdk.xdr.asset_code4.**AssetCode4**(*asset_code4*)
XDR Source Code:

```
typedef opaque AssetCode4[4];
```

AssetType

class stellar_sdk.xdr.asset_type.**AssetType**(*value*)
XDR Source Code:

```
enum AssetType
{
    ASSET_TYPE_NATIVE = 0,
    ASSET_TYPE_CREDIT_ALPHANUM4 = 1,
    ASSET_TYPE_CREDIT_ALPHANUM12 = 2,
    ASSET_TYPE_POOL_SHARE = 3
};
```

Auth

class stellar_sdk.xdr.auth.**Auth**(*unused*)
XDR Source Code:

```
struct Auth
{
    // Empty message, just to confirm
    // establishment of MAC keys.
    int unused;
};
```

AuthCert

class stellar_sdk.xdr.auth_cert.**AuthCert**(*pubkey, expiration, sig*)
XDR Source Code:

```
struct AuthCert
{
    Curve25519Public pubkey;
    uint64 expiration;
    Signature sig;
};
```

AuthenticatedMessage

class stellar_sdk.xdr.authenticated_message.**AuthenticatedMessage**(*v*, *v0=None*)
 XDR Source Code:

```
union AuthenticatedMessage switch (uint32 v)
{
  case 0:
    struct
    {
      uint64 sequence;
      StellarMessage message;
      HmacSha256Mac mac;
    } v0;
};
```

AuthenticatedMessageV0

class stellar_sdk.xdr.authenticated_message_v0.**AuthenticatedMessageV0**(*sequence*, *message*, *mac*)
 XDR Source Code:

```
struct
{
  uint64 sequence;
  StellarMessage message;
  HmacSha256Mac mac;
}
```

BeginSponsoringFutureReservesOp

class stellar_sdk.xdr.begin_sponsoring_future_reserves_op.**BeginSponsoringFutureReservesOp**(*sponsored_id*)
 XDR Source Code:

```
struct BeginSponsoringFutureReservesOp
{
  AccountID sponsoredID;
};
```

BeginSponsoringFutureReservesResult

class stellar_sdk.xdr.begin_sponsoring_future_reserves_result.**BeginSponsoringFutureReservesResult**(*code*)
 XDR Source Code:

```
union BeginSponsoringFutureReservesResult switch (
  BeginSponsoringFutureReservesResultCode code)
{
  case BEGIN_SPONSORING_FUTURE_RESERVES_SUCCESS:
    void;
  default:
    void;
};
```

BeginSponsoringFutureReservesResultCode

`class stellar_sdk.xdr.begin_sponsoring_future_reserves_result_code.BeginSponsoringFutureReservesResultCode`
XDR Source Code:

```
enum BeginSponsoringFutureReservesResultCode
{
    // codes considered as "success" for the operation
    BEGIN_SPONSORING_FUTURE_RESERVES_SUCCESS = 0,

    // codes considered as "failure" for the operation
    BEGIN_SPONSORING_FUTURE_RESERVES_MALFORMED = -1,
    BEGIN_SPONSORING_FUTURE_RESERVES_ALREADY_SPONSORED = -2,
    BEGIN_SPONSORING_FUTURE_RESERVES_RECURSIVE = -3
};
```

Boolean

`class stellar_sdk.xdr.base.Boolean(value)`

BucketEntry

`class stellar_sdk.xdr.bucket_entry.BucketEntry(type, live_entry=None, dead_entry=None, meta_entry=None)`

XDR Source Code:

```
union BucketEntry switch (BucketEntryType type)
{
    case LIVEENTRY:
    case INITENTRY:
        LedgerEntry liveEntry;

    case DEADENTRY:
        LedgerKey deadEntry;
    case METAENTRY:
        BucketMetadata metaEntry;
};
```

BucketEntryType

`class stellar_sdk.xdr.bucket_entry_type.BucketEntryType(value)`

XDR Source Code:

```
enum BucketEntryType
{
    METAENTRY =
        -1, // At-and-after protocol 11: bucket metadata, should come first.
    LIVEENTRY = 0, // Before protocol 11: created-or-updated;
        // At-and-after protocol 11: only updated.
    DEADENTRY = 1,
```

(continues on next page)

(continued from previous page)

```
INITENTRY = 2 // At-and-after protocol 11: only created.
};
```

BucketMetadata

`class stellar_sdk.xdr.bucket_metadata.BucketMetadata(ledger_version, ext)`

XDR Source Code:

```
struct BucketMetadata
{
    // Indicates the protocol version used to create / merge this bucket.
    uint32 ledgerVersion;

    // reserved for future use
    union switch (int v)
    {
        case 0:
            void;
    }
    ext;
};
```

BucketMetadataExt

`class stellar_sdk.xdr.bucket_metadata_ext.BucketMetadataExt(v)`

XDR Source Code:

```
union switch (int v)
{
    case 0:
        void;
}
```

BumpSequenceOp

`class stellar_sdk.xdr.bump_sequence_op.BumpSequenceOp(bump_to)`

XDR Source Code:

```
struct BumpSequenceOp
{
    SequenceNumber bumpTo;
};
```

BumpSequenceResult

class stellar_sdk.xdr.bump_sequence_result.**BumpSequenceResult**(*code*)
XDR Source Code:

```
union BumpSequenceResult switch (BumpSequenceResultCode code)
{
  case BUMP_SEQUENCE_SUCCESS:
    void;
  default:
    void;
};
```

BumpSequenceResultCode

class stellar_sdk.xdr.bump_sequence_result_code.**BumpSequenceResultCode**(*value*)
XDR Source Code:

```
enum BumpSequenceResultCode
{
  // codes considered as "success" for the operation
  BUMP_SEQUENCE_SUCCESS = 0,
  // codes considered as "failure" for the operation
  BUMP_SEQUENCE_BAD_SEQ = -1 // `bumpTo` is not within bounds
};
```

ChangeTrustAsset

class stellar_sdk.xdr.change_trust_asset.**ChangeTrustAsset**(*type*, *alpha_num4=*None,
*alpha_num12=*None,
*liquidity_pool=*None)

XDR Source Code:

```
union ChangeTrustAsset switch (AssetType type)
{
  case ASSET_TYPE_NATIVE: // Not credit
    void;

  case ASSET_TYPE_CREDIT_ALPHANUM4:
    AlphaNum4 alphaNum4;

  case ASSET_TYPE_CREDIT_ALPHANUM12:
    AlphaNum12 alphaNum12;

  case ASSET_TYPE_POOL_SHARE:
    LiquidityPoolParameters liquidityPool;

  // add other asset types here in the future
};
```

ChangeTrustOp

`class stellar_sdk.xdr.change_trust_op.ChangeTrustOp(line, limit)`

XDR Source Code:

```

struct ChangeTrustOp
{
    ChangeTrustAsset line;

    // if limit is set to 0, deletes the trust line
    int64 limit;
};

```

ChangeTrustResult

`class stellar_sdk.xdr.change_trust_result.ChangeTrustResult(code)`

XDR Source Code:

```

union ChangeTrustResult switch (ChangeTrustResultCode code)
{
    case CHANGE_TRUST_SUCCESS:
        void;
    default:
        void;
};

```

ChangeTrustResultCode

`class stellar_sdk.xdr.change_trust_result_code.ChangeTrustResultCode(value)`

XDR Source Code:

```

enum ChangeTrustResultCode
{
    // codes considered as "success" for the operation
    CHANGE_TRUST_SUCCESS = 0,
    // codes considered as "failure" for the operation
    CHANGE_TRUST_MALFORMED = -1, // bad input
    CHANGE_TRUST_NO_ISSUER = -2, // could not find issuer
    CHANGE_TRUST_INVALID_LIMIT = -3, // cannot drop limit below balance
    // cannot create with a limit of 0
    CHANGE_TRUST_LOW_RESERVE =
        -4, // not enough funds to create a new trust line,
    CHANGE_TRUST_SELF_NOT_ALLOWED = -5, // trusting self is not allowed
    CHANGE_TRUST_TRUST_LINE_MISSING = -6, // Asset trustline is missing for pool
    CHANGE_TRUST_CANNOT_DELETE = -7, // Asset trustline is still referenced in a
    ↪pool
    CHANGE_TRUST_NOT_AUTH_MAINTAIN_LIABILITIES = -8 // Asset trustline is
    ↪deauthorized
};

```

ClaimAtom

class stellar_sdk.xdr.claim_atom.ClaimAtom(*type*, *v0=None*, *order_book=None*, *liquidity_pool=None*)
XDR Source Code:

```
union ClaimAtom switch (ClaimAtomType type)
{
  case CLAIM_ATOM_TYPE_V0:
    ClaimOfferAtomV0 v0;
  case CLAIM_ATOM_TYPE_ORDER_BOOK:
    ClaimOfferAtom orderBook;
  case CLAIM_ATOM_TYPE_LIQUIDITY_POOL:
    ClaimLiquidityAtom liquidityPool;
};
```

ClaimAtomType

class stellar_sdk.xdr.claim_atom_type.ClaimAtomType(*value*)
XDR Source Code:

```
enum ClaimAtomType
{
  CLAIM_ATOM_TYPE_V0 = 0,
  CLAIM_ATOM_TYPE_ORDER_BOOK = 1,
  CLAIM_ATOM_TYPE_LIQUIDITY_POOL = 2
};
```

ClaimClaimableBalanceOp

class stellar_sdk.xdr.claim_claimable_balance_op.ClaimClaimableBalanceOp(*balance_id*)
XDR Source Code:

```
struct ClaimClaimableBalanceOp
{
  ClaimableBalanceID balanceID;
};
```

ClaimClaimableBalanceResult

class stellar_sdk.xdr.claim_claimable_balance_result.ClaimClaimableBalanceResult(*code*)
XDR Source Code:

```
union ClaimClaimableBalanceResult switch (ClaimClaimableBalanceResultCode code)
{
  case CLAIM_CLAIMABLE_BALANCE_SUCCESS:
    void;
  default:
    void;
};
```


ClaimClaimableBalanceResultCode

class stellar_sdk.xdr.claim_claimable_balance_result_code.ClaimClaimableBalanceResultCode(*value*)
XDR Source Code:

```
enum ClaimClaimableBalanceResultCode
{
    CLAIM_CLAIMABLE_BALANCE_SUCCESS = 0,
    CLAIM_CLAIMABLE_BALANCE_DOES_NOT_EXIST = -1,
    CLAIM_CLAIMABLE_BALANCE_CANNOT_CLAIM = -2,
    CLAIM_CLAIMABLE_BALANCE_LINE_FULL = -3,
    CLAIM_CLAIMABLE_BALANCE_NO_TRUST = -4,
    CLAIM_CLAIMABLE_BALANCE_NOT_AUTHORIZED = -5
};
```

ClaimLiquidityAtom

class stellar_sdk.xdr.claim_liquidity_atom.ClaimLiquidityAtom(*liquidity_pool_id*, *asset_sold*,
amount_sold, *asset_bought*,
amount_bought)

XDR Source Code:

```
struct ClaimLiquidityAtom
{
    PoolID liquidityPoolID;

    // amount and asset taken from the pool
    Asset assetSold;
    int64 amountSold;

    // amount and asset sent to the pool
    Asset assetBought;
    int64 amountBought;
};
```

ClaimOfferAtom

class stellar_sdk.xdr.claim_offer_atom.ClaimOfferAtom(*seller_id*, *offer_id*, *asset_sold*, *amount_sold*,
asset_bought, *amount_bought*)

XDR Source Code:

```
struct ClaimOfferAtom
{
    // emitted to identify the offer
    AccountID sellerID; // Account that owns the offer
    int64 offerID;

    // amount and asset taken from the owner
    Asset assetSold;
    int64 amountSold;
};
```

(continues on next page)

(continued from previous page)

```

// amount and asset sent to the owner
Asset assetBought;
int64 amountBought;
};

```

ClaimOfferAtomV0

```

class stellar_sdk.xdr.claim_offer_atom_v0.ClaimOfferAtomV0(seller_ed25519, offer_id, asset_sold,
                                                           amount_sold, asset_bought,
                                                           amount_bought)

```

XDR Source Code:

```

struct ClaimOfferAtomV0
{
    // emitted to identify the offer
    uint256 sellerEd25519; // Account that owns the offer
    int64 offerID;

    // amount and asset taken from the owner
    Asset assetSold;
    int64 amountSold;

    // amount and asset sent to the owner
    Asset assetBought;
    int64 amountBought;
};

```

ClaimPredicate

```

class stellar_sdk.xdr.claim_predicate.ClaimPredicate(type, and_predicates=None,
                                                    or_predicates=None, not_predicate=None,
                                                    abs_before=None, rel_before=None)

```

XDR Source Code:

```

union ClaimPredicate switch (ClaimPredicateType type)
{
    case CLAIM_PREDICATE_UNCONDITIONAL:
        void;
    case CLAIM_PREDICATE_AND:
        ClaimPredicate andPredicates<2>;
    case CLAIM_PREDICATE_OR:
        ClaimPredicate orPredicates<2>;
    case CLAIM_PREDICATE_NOT:
        ClaimPredicate* notPredicate;
    case CLAIM_PREDICATE_BEFORE_ABSOLUTE_TIME:
        int64 absBefore; // Predicate will be true if closeTime < absBefore
    case CLAIM_PREDICATE_BEFORE_RELATIVE_TIME:
        int64 relBefore; // Seconds since closeTime of the ledger in which the

```

(continues on next page)

(continued from previous page)

```

};
        // ClaimableBalanceEntry was created
};

```

ClaimPredicateType

`class stellar_sdk.xdr.claim_predicate_type.ClaimPredicateType(value)`

XDR Source Code:

```

enum ClaimPredicateType
{
    CLAIM_PREDICATE_UNCONDITIONAL = 0,
    CLAIM_PREDICATE_AND = 1,
    CLAIM_PREDICATE_OR = 2,
    CLAIM_PREDICATE_NOT = 3,
    CLAIM_PREDICATE_BEFORE_ABSOLUTE_TIME = 4,
    CLAIM_PREDICATE_BEFORE_RELATIVE_TIME = 5
};

```

ClaimableBalanceEntry

`class stellar_sdk.xdr.claimable_balance_entry.ClaimableBalanceEntry(balance_id, claimants, asset, amount, ext)`

XDR Source Code:

```

struct ClaimableBalanceEntry
{
    // Unique identifier for this ClaimableBalanceEntry
    ClaimableBalanceID balanceID;

    // List of claimants with associated predicate
    Claimant claimants<10>;

    // Any asset including native
    Asset asset;

    // Amount of asset
    int64 amount;

    // reserved for future use
    union switch (int v)
    {
        case 0:
            void;
        case 1:
            ClaimableBalanceEntryExtensionV1 v1;
    }
    ext;
};

```

ClaimableBalanceEntryExt

class stellar_sdk.xdr.claimable_balance_entry_ext.ClaimableBalanceEntryExt(*v*, *vI=None*)
XDR Source Code:

```
union switch (int v)
{
  case 0:
    void;
  case 1:
    ClaimableBalanceEntryExtensionV1 v1;
}
```

ClaimableBalanceEntryExtensionV1

class stellar_sdk.xdr.claimable_balance_entry_extension_v1.ClaimableBalanceEntryExtensionV1(*ext*,
flags)

XDR Source Code:

```
struct ClaimableBalanceEntryExtensionV1
{
  union switch (int v)
  {
    case 0:
      void;
  }
  ext;

  uint32 flags; // see ClaimableBalanceFlags
};
```

ClaimableBalanceEntryExtensionV1Ext

class stellar_sdk.xdr.claimable_balance_entry_extension_v1_ext.ClaimableBalanceEntryExtensionV1Ext(*v*)
XDR Source Code:

```
union switch (int v)
{
  case 0:
    void;
}
```

ClaimableBalanceFlags

class stellar_sdk.xdr.claimable_balance_flags.ClaimableBalanceFlags(*value*)

XDR Source Code:

```
enum ClaimableBalanceFlags
{
    // If set, the issuer account of the asset held by the claimable balance may
    // clawback the claimable balance
    CLAIMABLE_BALANCE_CLAWBACK_ENABLED_FLAG = 0x1
};
```

ClaimableBalanceID

class stellar_sdk.xdr.claimable_balance_id.ClaimableBalanceID(*type*, *v0=None*)

XDR Source Code:

```
union ClaimableBalanceID switch (ClaimableBalanceIDType type)
{
    case CLAIMABLE_BALANCE_ID_TYPE_V0:
        Hash v0;
};
```

ClaimableBalanceIDType

class stellar_sdk.xdr.claimable_balance_id_type.ClaimableBalanceIDType(*value*)

XDR Source Code:

```
enum ClaimableBalanceIDType
{
    CLAIMABLE_BALANCE_ID_TYPE_V0 = 0
};
```

Claimant

class stellar_sdk.xdr.claimant.Claimant(*type*, *v0=None*)

XDR Source Code:

```
union Claimant switch (ClaimantType type)
{
    case CLAIMANT_TYPE_V0:
        struct
        {
            AccountID destination; // The account that can use this condition
            ClaimPredicate predicate; // Claimable if predicate is true
        } v0;
};
```

ClaimantType

class stellar_sdk.xdr.claimant_type.ClaimantType(*value*)

XDR Source Code:

```
enum ClaimantType
{
    CLAIMANT_TYPE_V0 = 0
};
```

ClaimantV0

class stellar_sdk.xdr.claimant_v0.ClaimantV0(*destination*, *predicate*)

XDR Source Code:

```
struct
{
    AccountID destination; // The account that can use this condition
    ClaimPredicate predicate; // Claimable if predicate is true
}
```

ClawbackClaimableBalanceOp

class stellar_sdk.xdr.clawback_claimable_balance_op.ClawbackClaimableBalanceOp(*balance_id*)

XDR Source Code:

```
struct ClawbackClaimableBalanceOp
{
    ClaimableBalanceID balanceID;
};
```

ClawbackClaimableBalanceResult

class stellar_sdk.xdr.clawback_claimable_balance_result.ClawbackClaimableBalanceResult(*code*)

XDR Source Code:

```
union ClawbackClaimableBalanceResult switch (
    ClawbackClaimableBalanceResultCode code)
{
    case CLAWBACK_CLAIMABLE_BALANCE_SUCCESS:
        void;
    default:
        void;
};
```

ClawbackClaimableBalanceResultCode

class stellar_sdk.xdr.clawback_claimable_balance_result_code.**ClawbackClaimableBalanceResultCode**(*value*)
XDR Source Code:

```
enum ClawbackClaimableBalanceResultCode
{
    // codes considered as "success" for the operation
    CLAWBACK_CLAIMABLE_BALANCE_SUCCESS = 0,

    // codes considered as "failure" for the operation
    CLAWBACK_CLAIMABLE_BALANCE_DOES_NOT_EXIST = -1,
    CLAWBACK_CLAIMABLE_BALANCE_NOT_ISSUER = -2,
    CLAWBACK_CLAIMABLE_BALANCE_NOT_CLAWBACK_ENABLED = -3
};
```

ClawbackOp

class stellar_sdk.xdr.clawback_op.**ClawbackOp**(*asset, from_, amount*)
XDR Source Code:

```
struct ClawbackOp
{
    Asset asset;
    MuxedAccount from_;
    int64 amount;
};
```

ClawbackResult

class stellar_sdk.xdr.clawback_result.**ClawbackResult**(*code*)
XDR Source Code:

```
union ClawbackResult switch (ClawbackResultCode code)
{
    case CLAWBACK_SUCCESS:
        void;
    default:
        void;
};
```

ClawbackResultCode

class stellar_sdk.xdr.clawback_result_code.**ClawbackResultCode**(*value*)
XDR Source Code:

```
enum ClawbackResultCode
{
    // codes considered as "success" for the operation
    CLAWBACK_SUCCESS = 0,
```

(continues on next page)

(continued from previous page)

```

// codes considered as "failure" for the operation
CLAWBACK_MALFORMED = -1,
CLAWBACK_NOT_CLAWBACK_ENABLED = -2,
CLAWBACK_NO_TRUST = -3,
CLAWBACK_UNDERFUNDED = -4
};

```

CreateAccountOp

`class stellar_sdk.xdr.create_account_op.CreateAccountOp(destination, starting_balance)`

XDR Source Code:

```

struct CreateAccountOp
{
    AccountID destination; // account to create
    int64 startingBalance; // amount they end up with
};

```

CreateAccountResult

`class stellar_sdk.xdr.create_account_result.CreateAccountResult(code)`

XDR Source Code:

```

union CreateAccountResult switch (CreateAccountResultCode code)
{
    case CREATE_ACCOUNT_SUCCESS:
        void;
    default:
        void;
};

```

CreateAccountResultCode

`class stellar_sdk.xdr.create_account_result_code.CreateAccountResultCode(value)`

XDR Source Code:

```

enum CreateAccountResultCode
{
    // codes considered as "success" for the operation
    CREATE_ACCOUNT_SUCCESS = 0, // account was created

    // codes considered as "failure" for the operation
    CREATE_ACCOUNT_MALFORMED = -1, // invalid destination
    CREATE_ACCOUNT_UNDERFUNDED = -2, // not enough funds in source account
    CREATE_ACCOUNT_LOW_RESERVE =
        -3, // would create an account below the min reserve
    CREATE_ACCOUNT_ALREADY_EXIST = -4 // account already exists
};

```


CreateClaimableBalanceOp

class stellar_sdk.xdr.create_claimable_balance_op.**CreateClaimableBalanceOp**(*asset, amount, claimants*)

XDR Source Code:

```
struct CreateClaimableBalanceOp
{
    Asset asset;
    int64 amount;
    Claimant claimants<10>;
};
```

CreateClaimableBalanceResult

class stellar_sdk.xdr.create_claimable_balance_result.**CreateClaimableBalanceResult**(*code, balance_id=None*)

XDR Source Code:

```
union CreateClaimableBalanceResult switch (
    CreateClaimableBalanceResultCode code)
{
    case CREATE_CLAIMABLE_BALANCE_SUCCESS:
        ClaimableBalanceID balanceID;
    default:
        void;
};
```

CreateClaimableBalanceResultCode

class stellar_sdk.xdr.create_claimable_balance_result_code.**CreateClaimableBalanceResultCode**(*value*)

XDR Source Code:

```
enum CreateClaimableBalanceResultCode
{
    CREATE_CLAIMABLE_BALANCE_SUCCESS = 0,
    CREATE_CLAIMABLE_BALANCE_MALFORMED = -1,
    CREATE_CLAIMABLE_BALANCE_LOW_RESERVE = -2,
    CREATE_CLAIMABLE_BALANCE_NO_TRUST = -3,
    CREATE_CLAIMABLE_BALANCE_NOT_AUTHORIZED = -4,
    CREATE_CLAIMABLE_BALANCE_UNDERFUNDED = -5
};
```

CreatePassiveSellOfferOp

class stellar_sdk.xdr.create_passive_sell_offer_op.**CreatePassiveSellOfferOp**(*selling, buying, amount, price*)

XDR Source Code:

```
struct CreatePassiveSellOfferOp
{
    Asset selling; // A
    Asset buying; // B
    int64 amount; // amount taker gets
    Price price; // cost of A in terms of B
};
```

CryptoKeyType

class stellar_sdk.xdr.crypto_key_type.**CryptoKeyType**(*value*)

XDR Source Code:

```
enum CryptoKeyType
{
    KEY_TYPE_ED25519 = 0,
    KEY_TYPE_PRE_AUTH_TX = 1,
    KEY_TYPE_HASH_X = 2,
    // MUXED enum values for supported type are derived from the enum values
    // above by ORing them with 0x100
    KEY_TYPE_MUXED_ED25519 = 0x100
};
```

Curve25519Public

class stellar_sdk.xdr.curve25519_public.**Curve25519Public**(*key*)

XDR Source Code:

```
struct Curve25519Public
{
    opaque key[32];
};
```

Curve25519Secret

class stellar_sdk.xdr.curve25519_secret.**Curve25519Secret**(*key*)

XDR Source Code:

```
struct Curve25519Secret
{
    opaque key[32];
};
```

DataEntry

class stellar_sdk.xdr.data_entry.**DataEntry**(*account_id, data_name, data_value, ext*)

XDR Source Code:

```

struct DataEntry
{
    AccountID accountID; // account this data belongs to
    string64 dataName;
    DataValue dataValue;

    // reserved for future use
    union switch (int v)
    {
        case 0:
            void;
    }
    ext;
};

```

DataEntryExt

class stellar_sdk.xdr.data_entry_ext.**DataEntryExt**(*v*)

XDR Source Code:

```

union switch (int v)
{
    case 0:
        void;
}

```

DataValue

class stellar_sdk.xdr.data_value.**DataValue**(*data_value*)

XDR Source Code:

```

typedef opaque DataValue<64>;

```

DecoratedSignature

class stellar_sdk.xdr.decorated_signature.**DecoratedSignature**(*hint, signature*)

XDR Source Code:

```

struct DecoratedSignature
{
    SignatureHint hint; // last 4 bytes of the public key, used as a hint
    Signature signature; // actual signature
};

```

DontHave

class stellar_sdk.xdr.dont_have.**DontHave**(*type*, *req_hash*)
XDR Source Code:

```
struct DontHave
{
    MessageType type;
    uint256 reqHash;
};
```

EncryptedBody

class stellar_sdk.xdr.encrypted_body.**EncryptedBody**(*encrypted_body*)
XDR Source Code:

```
typedef opaque EncryptedBody<64000>;
```

EndSponsoringFutureReservesResult

class stellar_sdk.xdr.end_sponsoring_future_reserves_result.**EndSponsoringFutureReservesResult**(*code*)
XDR Source Code:

```
union EndSponsoringFutureReservesResult switch (
    EndSponsoringFutureReservesResultCode code)
{
    case END_SPONSORING_FUTURE_RESERVES_SUCCESS:
        void;
    default:
        void;
};
```

EndSponsoringFutureReservesResultCode

class stellar_sdk.xdr.end_sponsoring_future_reserves_result_code.**EndSponsoringFutureReservesResultCode**(*code*)
XDR Source Code:

```
enum EndSponsoringFutureReservesResultCode
{
    // codes considered as "success" for the operation
    END_SPONSORING_FUTURE_RESERVES_SUCCESS = 0,

    // codes considered as "failure" for the operation
    END_SPONSORING_FUTURE_RESERVES_NOT_SPONSORED = -1
};
```

EnvelopeType

class stellar_sdk.xdr.envelope_type.**EnvelopeType**(*value*)

XDR Source Code:

```
enum EnvelopeType
{
    ENVELOPE_TYPE_TX_V0 = 0,
    ENVELOPE_TYPE_SCP = 1,
    ENVELOPE_TYPE_TX = 2,
    ENVELOPE_TYPE_AUTH = 3,
    ENVELOPE_TYPE_SCPVALUE = 4,
    ENVELOPE_TYPE_TX_FEE_BUMP = 5,
    ENVELOPE_TYPE_OP_ID = 6
};
```

Error

class stellar_sdk.xdr.error.**Error**(*code, msg*)

XDR Source Code:

```
struct Error
{
    ErrorCode code;
    string msg<100>;
};
```

ErrorCode

class stellar_sdk.xdr.error_code.**ErrorCode**(*value*)

XDR Source Code:

```
enum ErrorCode
{
    ERR_MISC = 0, // Unspecific error
    ERR_DATA = 1, // Malformed data
    ERR_CONF = 2, // Misconfiguration error
    ERR_AUTH = 3, // Authentication failure
    ERR_LOAD = 4 // System overloaded
};
```

FeeBumpTransaction

class stellar_sdk.xdr.fee_bump_transaction.**FeeBumpTransaction**(*fee_source, fee, inner_tx, ext*)

XDR Source Code:

```
struct FeeBumpTransaction
{
    MuxedAccount feeSource;
    int64 fee;
```

(continues on next page)

(continued from previous page)

```

union switch (EnvelopeType type)
{
  case ENVELOPE_TYPE_TX:
    TransactionV1Envelope v1;
}
innerTx;
union switch (int v)
{
  case 0:
    void;
}
ext;
};

```

FeeBumpTransactionEnvelope

```
class stellar_sdk.xdr.fee_bump_transaction_envelope.FeeBumpTransactionEnvelope(tx,
                                                                              signatures)
```

XDR Source Code:

```

struct FeeBumpTransactionEnvelope
{
  FeeBumpTransaction tx;
  /* Each decorated signature is a signature over the SHA256 hash of
   * a TransactionSignaturePayload */
  DecoratedSignature signatures<20>;
};

```

FeeBumpTransactionExt

```
class stellar_sdk.xdr.fee_bump_transaction_ext.FeeBumpTransactionExt(v)
```

XDR Source Code:

```

union switch (int v)
{
  case 0:
    void;
}

```

FeeBumpTransactionInnerTx

```
class stellar_sdk.xdr.fee_bump_transaction_inner_tx.FeeBumpTransactionInnerTx(type,
                                                                                vI=None)
```

XDR Source Code:

```

union switch (EnvelopeType type)
{
  case ENVELOPE_TYPE_TX:

```

(continues on next page)

(continued from previous page)

```
TransactionV1Envelope v1;
}
```

Hash

class stellar_sdk.xdr.hash.**Hash**(*hash*)

XDR Source Code:

```
typedef opaque Hash[32];
```

Hello

class stellar_sdk.xdr.hello.**Hello**(*ledger_version, overlay_version, overlay_min_version, network_id, version_str, listening_port, peer_id, cert, nonce*)

XDR Source Code:

```
struct Hello
{
    uint32 ledgerVersion;
    uint32 overlayVersion;
    uint32 overlayMinVersion;
    Hash networkID;
    string versionStr<100>;
    int listeningPort;
    NodeID peerID;
    AuthCert cert;
    uint256 nonce;
};
```

HmacSha256Key

class stellar_sdk.xdr.hmac_sha256_key.**HmacSha256Key**(*key*)

XDR Source Code:

```
struct HmacSha256Key
{
    opaque key[32];
};
```

HmacSha256Mac

class stellar_sdk.xdr.hmac_sha256_mac.**HmacSha256Mac**(*mac*)

XDR Source Code:

```
struct HmacSha256Mac
{
    opaque mac[32];
};
```

Hyper

`class stellar_sdk.xdr.base.Hyper(value)`

IPAddrType

`class stellar_sdk.xdr.ip_addr_type.IPAddrType(value)`

XDR Source Code:

```
enum IPAddrType
{
    IPv4 = 0,
    IPv6 = 1
};
```

InflationPayout

`class stellar_sdk.xdr.inflation_payout.InflationPayout(destination, amount)`

XDR Source Code:

```
struct InflationPayout // or use PaymentResultAtom to limit types?
{
    AccountID destination;
    int64 amount;
};
```

InflationResult

`class stellar_sdk.xdr.inflation_result.InflationResult(code, payouts=None)`

XDR Source Code:

```
union InflationResult switch (InflationResultCode code)
{
    case INFLATION_SUCCESS:
        InflationPayout payouts<>;
    default:
        void;
};
```

InflationResultCode

`class stellar_sdk.xdr.inflation_result_code.InflationResultCode(value)`

XDR Source Code:

```
enum InflationResultCode
{
    // codes considered as "success" for the operation
    INFLATION_SUCCESS = 0,
    // codes considered as "failure" for the operation
```

(continues on next page)

(continued from previous page)

```

INFLATION_NOT_TIME = -1
};

```

InnerTransactionResult

`class stellar_sdk.xdr.inner_transaction_result.InnerTransactionResult(fee_charged, result, ext)`
 XDR Source Code:

```

struct InnerTransactionResult
{
    // Always 0. Here for binary compatibility.
    int64 feeCharged;

    union switch (TransactionResultCode code)
    {
        // txFEE_BUMP_INNER_SUCCESS is not included
        case txSUCCESS:
        case txFAILED:
            OperationResult results<>;
        case txTOO_EARLY:
        case txTOO_LATE:
        case txMISSING_OPERATION:
        case txBAD_SEQ:
        case txBAD_AUTH:
        case txINSUFFICIENT_BALANCE:
        case txNO_ACCOUNT:
        case txINSUFFICIENT_FEE:
        case txBAD_AUTH_EXTRA:
        case txINTERNAL_ERROR:
        case txNOT_SUPPORTED:
        // txFEE_BUMP_INNER_FAILED is not included
        case txBAD_SPONSORSHIP:
            void;
    }
    result;

    // reserved for future use
    union switch (int v)
    {
        case 0:
            void;
    }
    ext;
};

```

InnerTransactionResultExt

`class stellar_sdk.xdr.inner_transaction_result_ext.InnerTransactionResultExt(v)`
XDR Source Code:

```
union switch (int v)
{
  case 0:
    void;
}
```

InnerTransactionResultPair

`class stellar_sdk.xdr.inner_transaction_result_pair.InnerTransactionResultPair(transaction_hash, result)`

XDR Source Code:

```
struct InnerTransactionResultPair
{
  Hash transactionHash; // hash of the inner transaction
  InnerTransactionResult result; // result for the inner transaction
};
```

InnerTransactionResultResult

`class stellar_sdk.xdr.inner_transaction_result_result.InnerTransactionResultResult(code, re-sults=None)`

XDR Source Code:

```
union switch (TransactionResultCode code)
{
  // txFEE_BUMP_INNER_SUCCESS is not included
  case txSUCCESS:
  case txFAILED:
    OperationResult results<>;
  case txTOO_EARLY:
  case txTOO_LATE:
  case txMISSING_OPERATION:
  case txBAD_SEQ:
  case txBAD_AUTH:
  case txINSUFFICIENT_BALANCE:
  case txNO_ACCOUNT:
  case txINSUFFICIENT_FEE:
  case txBAD_AUTH_EXTRA:
  case txINTERNAL_ERROR:
  case txNOT_SUPPORTED:
  // txFEE_BUMP_INNER_FAILED is not included
  case txBAD_SPONSORSHIP:
    void;
}
```

Int32

`class stellar_sdk.xdr.int32.Int32(int32)`
 XDR Source Code:

```
typedef int int32;
```

Int64

`class stellar_sdk.xdr.int64.Int64(int64)`
 XDR Source Code:

```
typedef hyper int64;
```

Integer

`class stellar_sdk.xdr.base.Integer(value)`

LedgerCloseMeta

`class stellar_sdk.xdr.ledger_close_meta.LedgerCloseMeta(v, v0=None)`
 XDR Source Code:

```
union LedgerCloseMeta switch (int v)
{
  case 0:
    LedgerCloseMetaV0 v0;
};
```

LedgerCloseMetaV0

`class stellar_sdk.xdr.ledger_close_meta_v0.LedgerCloseMetaV0(ledger_header, tx_set, tx_processing, upgrades_processing, scp_info)`

XDR Source Code:

```
struct LedgerCloseMetaV0
{
  LedgerHeaderHistoryEntry ledgerHeader;
  // NB: txSet is sorted in "Hash order"
  TransactionSet txSet;

  // NB: transactions are sorted in apply order here
  // fees for all transactions are processed first
  // followed by applying transactions
  TransactionResultMeta txProcessing<>;

  // upgrades are applied last
  UpgradeEntryMeta upgradesProcessing<>;

  // other misc information attached to the ledger close
```

(continues on next page)

(continued from previous page)

```

SCPHistoryEntry scpInfo<>;
};

```

LedgerCloseValueSignature

```

class stellar_sdk.xdr.ledger_close_value_signature.LedgerCloseValueSignature(node_id,
                                                                           signature)

```

XDR Source Code:

```

struct LedgerCloseValueSignature
{
    NodeID nodeID;          // which node introduced the value
    Signature signature;    // nodeID's signature
};

```

LedgerEntry

```

class stellar_sdk.xdr.ledger_entry.LedgerEntry(last_modified_ledger_seq, data, ext)

```

XDR Source Code:

```

struct LedgerEntry
{
    uint32 lastModifiedLedgerSeq; // ledger the LedgerEntry was last changed

    union switch (LedgerEntryType type)
    {
        case ACCOUNT:
            AccountEntry account;
        case TRUSTLINE:
            TrustLineEntry trustLine;
        case OFFER:
            OfferEntry offer;
        case DATA:
            DataEntry data;
        case CLAIMABLE_BALANCE:
            ClaimableBalanceEntry claimableBalance;
        case LIQUIDITY_POOL:
            LiquidityPoolEntry liquidityPool;
    }
    data;

    // reserved for future use
    union switch (int v)
    {
        case 0:
            void;
        case 1:
            LedgerEntryExtensionV1 v1;
    }
}

```

(continues on next page)

(continued from previous page)

```

    ext;
};

```

LedgerEntryChange

class stellar_sdk.xdr.ledger_entry_change.LedgerEntryChange(*type*, *created=None*, *updated=None*, *removed=None*, *state=None*)

XDR Source Code:

```

union LedgerEntryChange switch (LedgerEntryChangeType type)
{
case LEDGER_ENTRY_CREATED:
    LedgerEntry created;
case LEDGER_ENTRY_UPDATED:
    LedgerEntry updated;
case LEDGER_ENTRY_REMOVED:
    LedgerKey removed;
case LEDGER_ENTRY_STATE:
    LedgerEntry state;
};

```

LedgerEntryChangeType

class stellar_sdk.xdr.ledger_entry_change_type.LedgerEntryChangeType(*value*)

XDR Source Code:

```

enum LedgerEntryChangeType
{
    LEDGER_ENTRY_CREATED = 0, // entry was added to the ledger
    LEDGER_ENTRY_UPDATED = 1, // entry was modified in the ledger
    LEDGER_ENTRY_REMOVED = 2, // entry was removed from the ledger
    LEDGER_ENTRY_STATE = 3    // value of the entry
};

```

LedgerEntryChanges

class stellar_sdk.xdr.ledger_entry_changes.LedgerEntryChanges(*ledger_entry_changes*)

XDR Source Code:

```

typedef LedgerEntryChange LedgerEntryChanges<>;

```

LedgerEntryData

```
class stellar_sdk.xdr.ledger_entry_data.LedgerEntryData(type, account=None, trust_line=None,  
                                                       offer=None, data=None,  
                                                       claimable_balance=None,  
                                                       liquidity_pool=None)
```

XDR Source Code:

```
union switch (LedgerEntryType type)  
{  
  case ACCOUNT:  
    AccountEntry account;  
  case TRUSTLINE:  
    TrustLineEntry trustLine;  
  case OFFER:  
    OfferEntry offer;  
  case DATA:  
    DataEntry data;  
  case CLAIMABLE_BALANCE:  
    ClaimableBalanceEntry claimableBalance;  
  case LIQUIDITY_POOL:  
    LiquidityPoolEntry liquidityPool;  
}
```

LedgerEntryExt

```
class stellar_sdk.xdr.ledger_entry_ext.LedgerEntryExt(v, v1=None)
```

XDR Source Code:

```
union switch (int v)  
{  
  case 0:  
    void;  
  case 1:  
    LedgerEntryExtensionV1 v1;  
}
```

LedgerEntryExtensionV1

```
class stellar_sdk.xdr.ledger_entry_extension_v1.LedgerEntryExtensionV1(sponsoring_id, ext)
```

XDR Source Code:

```
struct LedgerEntryExtensionV1  
{  
  SponsorshipDescriptor sponsoringID;  
  
  union switch (int v)  
  {  
    case 0:  
      void;  
  }  
}
```

(continues on next page)

(continued from previous page)

```

    ext;
};

```

LedgerEntryExtensionV1Ext

`class stellar_sdk.xdr.ledger_entry_extension_v1_ext.LedgerEntryExtensionV1Ext(v)`
XDR Source Code:

```

union switch (int v)
{
    case 0:
        void;
}

```

LedgerEntryType

`class stellar_sdk.xdr.ledger_entry_type.LedgerEntryType(value)`
XDR Source Code:

```

enum LedgerEntryType
{
    ACCOUNT = 0,
    TRUSTLINE = 1,
    OFFER = 2,
    DATA = 3,
    CLAIMABLE_BALANCE = 4,
    LIQUIDITY_POOL = 5
};

```

LedgerHeader

`class stellar_sdk.xdr.ledger_header.LedgerHeader(ledger_version, previous_ledger_hash, scp_value, tx_set_result_hash, bucket_list_hash, ledger_seq, total_coins, fee_pool, inflation_seq, id_pool, base_fee, base_reserve, max_tx_set_size, skip_list, ext)`

XDR Source Code:

```

struct LedgerHeader
{
    uint32 ledgerVersion; // the protocol version of the ledger
    Hash previousLedgerHash; // hash of the previous ledger header
    StellarValue scpValue; // what consensus agreed to
    Hash txSetResultHash; // the TransactionResultSet that led to this ledger
    Hash bucketListHash; // hash of the ledger state

    uint32 ledgerSeq; // sequence number of this ledger

    int64 totalCoins; // total number of stroops in existence.
}

```

(continues on next page)

(continued from previous page)

```

        // 10,000,000 stroops in 1 XLM

    int64 feePool;        // fees burned since last inflation run
    uint32 inflationSeq; // inflation sequence number

    uint64 idPool; // last used global ID, used for generating objects

    uint32 baseFee;      // base fee per operation in stroops
    uint32 baseReserve; // account base reserve in stroops

    uint32 maxTxSetSize; // maximum size a transaction set can be

    Hash skipList[4]; // hashes of ledgers in the past. allows you to jump back
                    // in time without walking the chain back ledger by ledger
                    // each slot contains the oldest ledger that is mod of
                    // either 50 5000 50000 or 500000 depending on index
                    // skipList[0] mod(50), skipList[1] mod(5000), etc

    // reserved for future use
    union switch (int v)
    {
    case 0:
        void;
    }
    ext;
};

```

LedgerHeaderExt

class stellar_sdk.xdr.ledger_header_ext.LedgerHeaderExt(*v*)

XDR Source Code:

```

union switch (int v)
{
case 0:
    void;
}

```

LedgerHeaderHistoryEntry

class stellar_sdk.xdr.ledger_header_history_entry.LedgerHeaderHistoryEntry(*hash, header, ext*)

XDR Source Code:

```

struct LedgerHeaderHistoryEntry
{
    Hash hash;
    LedgerHeader header;

    // reserved for future use
    union switch (int v)

```

(continues on next page)

(continued from previous page)

```

{
  case 0:
    void;
}
ext;
};

```

LedgerHeaderHistoryEntryExt

`class stellar_sdk.xdr.ledger_header_history_entry_ext.LedgerHeaderHistoryEntryExt(v)`
XDR Source Code:

```

union switch (int v)
{
  case 0:
    void;
}

```

LedgerKey

`class stellar_sdk.xdr.ledger_key.LedgerKey(type, account=None, trust_line=None, offer=None, data=None, claimable_balance=None, liquidity_pool=None)`

XDR Source Code:

```

union LedgerKey switch (LedgerEntryType type)
{
  case ACCOUNT:
    struct
    {
      AccountID accountID;
    } account;

  case TRUSTLINE:
    struct
    {
      AccountID accountID;
      TrustLineAsset asset;
    } trustLine;

  case OFFER:
    struct
    {
      AccountID sellerID;
      int64 offerID;
    } offer;

  case DATA:
    struct
    {

```

(continues on next page)

(continued from previous page)

```
        AccountID accountID;
        string64 dataName;
    } data;

    case CLAIMABLE_BALANCE:
        struct
        {
            ClaimableBalanceID balanceID;
        } claimableBalance;

    case LIQUIDITY_POOL:
        struct
        {
            PoolID liquidityPoolID;
        } liquidityPool;
};
```

LedgerKeyAccount

class stellar_sdk.xdr.ledger_key_account.LedgerKeyAccount(*account_id*)
XDR Source Code:

```
struct
{
    AccountID accountID;
}
```

LedgerKeyClaimableBalance

class stellar_sdk.xdr.ledger_key_claimable_balance.LedgerKeyClaimableBalance(*balance_id*)
XDR Source Code:

```
struct
{
    ClaimableBalanceID balanceID;
}
```

LedgerKeyData

class stellar_sdk.xdr.ledger_key_data.LedgerKeyData(*account_id*, *data_name*)
XDR Source Code:

```
struct
{
    AccountID accountID;
    string64 dataName;
}
```

LedgerKeyLiquidityPool

class stellar_sdk.xdr.ledger_key_liquidity_pool.**LedgerKeyLiquidityPool**(*liquidity_pool_id*)
XDR Source Code:

```
struct
{
    PoolID liquidityPoolID;
}
```

LedgerKeyOffer

class stellar_sdk.xdr.ledger_key_offer.**LedgerKeyOffer**(*seller_id, offer_id*)
XDR Source Code:

```
struct
{
    AccountID sellerID;
    int64 offerID;
}
```

LedgerKeyTrustLine

class stellar_sdk.xdr.ledger_key_trust_line.**LedgerKeyTrustLine**(*account_id, asset*)
XDR Source Code:

```
struct
{
    AccountID accountID;
    TrustLineAsset asset;
}
```

LedgerSCPMessages

class stellar_sdk.xdr.ledger_scp_messages.**LedgerSCPMessages**(*ledger_seq, messages*)
XDR Source Code:

```
struct LedgerSCPMessages
{
    uint32 ledgerSeq;
    SCPEnvelope messages<>;
};
```

LedgerUpgrade

```
class stellar_sdk.xdr.ledger_upgrade.LedgerUpgrade(type, new_ledger_version=None,  
                                                    new_base_fee=None,  
                                                    new_max_tx_set_size=None,  
                                                    new_base_reserve=None)
```

XDR Source Code:

```
union LedgerUpgrade switch (LedgerUpgradeType type)  
{  
  case LEDGER_UPGRADE_VERSION:  
    uint32 newLedgerVersion; // update ledgerVersion  
  case LEDGER_UPGRADE_BASE_FEE:  
    uint32 newBaseFee; // update baseFee  
  case LEDGER_UPGRADE_MAX_TX_SET_SIZE:  
    uint32 newMaxTxSetSize; // update maxTxSetSize  
  case LEDGER_UPGRADE_BASE_RESERVE:  
    uint32 newBaseReserve; // update baseReserve  
};
```

LedgerUpgradeType

```
class stellar_sdk.xdr.ledger_upgrade_type.LedgerUpgradeType(value)
```

XDR Source Code:

```
enum LedgerUpgradeType  
{  
  LEDGER_UPGRADE_VERSION = 1,  
  LEDGER_UPGRADE_BASE_FEE = 2,  
  LEDGER_UPGRADE_MAX_TX_SET_SIZE = 3,  
  LEDGER_UPGRADE_BASE_RESERVE = 4  
};
```

Liabilities

```
class stellar_sdk.xdr.liabilities.Liabilities(buying, selling)
```

XDR Source Code:

```
struct Liabilities  
{  
  int64 buying;  
  int64 selling;  
};
```

LiquidityPoolConstantProductParameters

```
class stellar_sdk.xdr.liquidity_pool_constant_product_parameters.LiquidityPoolConstantProductParameters
```

XDR Source Code:

```
struct LiquidityPoolConstantProductParameters
{
    Asset assetA; // assetA < assetB
    Asset assetB;
    int32 fee;    // Fee is in basis points, so the actual rate is (fee/100)%
};
```

LiquidityPoolDepositOp

```
class stellar_sdk.xdr.liquidity_pool_deposit_op.LiquidityPoolDepositOp(liquidity_pool_id,
                                                                    max_amount_a,
                                                                    max_amount_b,
                                                                    min_price, max_price)
```

XDR Source Code:

```
struct LiquidityPoolDepositOp
{
    PoolID liquidityPoolID;
    int64 maxAmountA;    // maximum amount of first asset to deposit
    int64 maxAmountB;    // maximum amount of second asset to deposit
    Price minPrice;      // minimum depositA/depositB
    Price maxPrice;      // maximum depositA/depositB
};
```

LiquidityPoolDepositResult

```
class stellar_sdk.xdr.liquidity_pool_deposit_result.LiquidityPoolDepositResult(code)
```

XDR Source Code:

```
union LiquidityPoolDepositResult switch (
    LiquidityPoolDepositResultCode code)
{
    case LIQUIDITY_POOL_DEPOSIT_SUCCESS:
        void;
    default:
        void;
};
```

LiquidityPoolDepositResultCode

`class stellar_sdk.xdr.liquidity_pool_deposit_result_code.LiquidityPoolDepositResultCode(value)`
XDR Source Code:

```
enum LiquidityPoolDepositResultCode
{
    // codes considered as "success" for the operation
    LIQUIDITY_POOL_DEPOSIT_SUCCESS = 0,

    // codes considered as "failure" for the operation
    LIQUIDITY_POOL_DEPOSIT_MALFORMED = -1, // bad input
    LIQUIDITY_POOL_DEPOSIT_NO_TRUST = -2, // no trust line for one of the
                                           // assets
    LIQUIDITY_POOL_DEPOSIT_NOT_AUTHORIZED = -3, // not authorized for one of the
                                                  // assets
    LIQUIDITY_POOL_DEPOSIT_UNDERFUNDED = -4, // not enough balance for one of
                                              // the assets
    LIQUIDITY_POOL_DEPOSIT_LINE_FULL = -5, // pool share trust line doesn't
                                             // have sufficient limit
    LIQUIDITY_POOL_DEPOSIT_BAD_PRICE = -6, // deposit price outside bounds
    LIQUIDITY_POOL_DEPOSIT_POOL_FULL = -7 // pool reserves are full
};
```

LiquidityPoolEntry

`class stellar_sdk.xdr.liquidity_pool_entry.LiquidityPoolEntry(liquidity_pool_id, body)`
XDR Source Code:

```
struct LiquidityPoolEntry
{
    PoolID liquidityPoolID;

    union switch (LiquidityPoolType type)
    {
        case LIQUIDITY_POOL_CONSTANT_PRODUCT:
            struct
            {
                LiquidityPoolConstantProductParameters params;

                int64 reserveA; // amount of A in the pool
                int64 reserveB; // amount of B in the pool
                int64 totalPoolShares; // total number of pool shares issued
                int64 poolSharesTrustLineCount; // number of trust lines for the
                ↪ associated pool shares
            } constantProduct;
    }
    body;
};
```

LiquidityPoolEntryBody

`class stellar_sdk.xdr.liquidity_pool_entry_body.LiquidityPoolEntryBody`(*type, constant_product=None*)

XDR Source Code:

```
union switch (LiquidityPoolType type)
{
  case LIQUIDITY_POOL_CONSTANT_PRODUCT:
    struct
    {
      LiquidityPoolConstantProductParameters params;

      int64 reserveA;          // amount of A in the pool
      int64 reserveB;          // amount of B in the pool
      int64 totalPoolShares; // total number of pool shares issued
      int64 poolSharesTrustLineCount; // number of trust lines for the
↪associated pool shares
    } constantProduct;
  }
}
```

LiquidityPoolEntryConstantProduct

`class stellar_sdk.xdr.liquidity_pool_entry_constant_product.LiquidityPoolEntryConstantProduct`(*params, reserve_a, reserve_b, total_pool_shares, pool_shares*)

XDR Source Code:

```
struct
{
  LiquidityPoolConstantProductParameters params;

  int64 reserveA;          // amount of A in the pool
  int64 reserveB;          // amount of B in the pool
  int64 totalPoolShares; // total number of pool shares issued
  int64 poolSharesTrustLineCount; // number of trust lines for the
↪associated pool shares
}
```

LiquidityPoolParameters

class stellar_sdk.xdr.liquidity_pool_parameters.LiquidityPoolParameters(*type*, *constant_product=None*)

XDR Source Code:

```
union LiquidityPoolParameters switch (LiquidityPoolType type)
{
  case LIQUIDITY_POOL_CONSTANT_PRODUCT:
    LiquidityPoolConstantProductParameters constantProduct;
};
```

LiquidityPoolType

class stellar_sdk.xdr.liquidity_pool_type.LiquidityPoolType(*value*)

XDR Source Code:

```
enum LiquidityPoolType
{
  LIQUIDITY_POOL_CONSTANT_PRODUCT = 0
};
```

LiquidityPoolWithdrawOp

class stellar_sdk.xdr.liquidity_pool_withdraw_op.LiquidityPoolWithdrawOp(*liquidity_pool_id*, *amount*, *min_amount_a*, *min_amount_b*)

XDR Source Code:

```
struct LiquidityPoolWithdrawOp
{
  PoolID liquidityPoolID;
  int64 amount;           // amount of pool shares to withdraw
  int64 minAmountA;      // minimum amount of first asset to withdraw
  int64 minAmountB;      // minimum amount of second asset to withdraw
};
```

LiquidityPoolWithdrawResult

class stellar_sdk.xdr.liquidity_pool_withdraw_result.LiquidityPoolWithdrawResult(*code*)

XDR Source Code:

```
union LiquidityPoolWithdrawResult switch (
  LiquidityPoolWithdrawResultCode code)
{
  case LIQUIDITY_POOL_WITHDRAW_SUCCESS:
    void;
  default:
    void;
};
```


LiquidityPoolWithdrawResultCode

`class stellar_sdk.xdr.liquidity_pool_withdraw_result_code.LiquidityPoolWithdrawResultCode(value)`
XDR Source Code:

```
enum LiquidityPoolWithdrawResultCode
{
    // codes considered as "success" for the operation
    LIQUIDITY_POOL_WITHDRAW_SUCCESS = 0,

    // codes considered as "failure" for the operation
    LIQUIDITY_POOL_WITHDRAW_MALFORMED = -1,    // bad input
    LIQUIDITY_POOL_WITHDRAW_NO_TRUST = -2,    // no trust line for one of the
                                                // assets
    LIQUIDITY_POOL_WITHDRAW_UNDERFUNDED = -3, // not enough balance of the
                                                // pool share
    LIQUIDITY_POOL_WITHDRAW_LINE_FULL = -4,    // would go above limit for one
                                                // of the assets
    LIQUIDITY_POOL_WITHDRAW_UNDER_MINIMUM = -5 // didn't withdraw enough
};
```

ManageBuyOfferOp

`class stellar_sdk.xdr.manage_buy_offer_op.ManageBuyOfferOp(selling, buying, buy_amount, price, offer_id)`

XDR Source Code:

```
struct ManageBuyOfferOp
{
    Asset selling;
    Asset buying;
    int64 buyAmount; // amount being bought. if set to 0, delete the offer
    Price price;    // price of thing being bought in terms of what you are
                  // selling

    // 0=create a new offer, otherwise edit an existing offer
    int64 offerID;
};
```

ManageBuyOfferResult

`class stellar_sdk.xdr.manage_buy_offer_result.ManageBuyOfferResult(code, success=None)`

XDR Source Code:

```
union ManageBuyOfferResult switch (ManageBuyOfferResultCode code)
{
    case MANAGE_BUY_OFFER_SUCCESS:
        ManageOfferSuccessResult success;
    default:
        void;
};
```

ManageBuyOfferResultCode

`class stellar_sdk.xdr.manage_buy_offer_result_code.ManageBuyOfferResultCode(value)`
XDR Source Code:

```
enum ManageBuyOfferResultCode
{
    // codes considered as "success" for the operation
    MANAGE_BUY_OFFER_SUCCESS = 0,

    // codes considered as "failure" for the operation
    MANAGE_BUY_OFFER_MALFORMED = -1, // generated offer would be invalid
    MANAGE_BUY_OFFER_SELL_NO_TRUST = -2, // no trust line for what we're selling
    MANAGE_BUY_OFFER_BUY_NO_TRUST = -3, // no trust line for what we're buying
    MANAGE_BUY_OFFER_SELL_NOT_AUTHORIZED = -4, // not authorized to sell
    MANAGE_BUY_OFFER_BUY_NOT_AUTHORIZED = -5, // not authorized to buy
    MANAGE_BUY_OFFER_LINE_FULL = -6, // can't receive more of what it's buying
    MANAGE_BUY_OFFER_UNDERFUNDED = -7, // doesn't hold what it's trying to sell
    MANAGE_BUY_OFFER_CROSS_SELF = -8, // would cross an offer from the same user
    MANAGE_BUY_OFFER_SELL_NO_ISSUER = -9, // no issuer for what we're selling
    MANAGE_BUY_OFFER_BUY_NO_ISSUER = -10, // no issuer for what we're buying

    // update errors
    MANAGE_BUY_OFFER_NOT_FOUND =
        -11, // offerID does not match an existing offer

    MANAGE_BUY_OFFER_LOW_RESERVE = -12 // not enough funds to create a new Offer
};
```

ManageDataOp

`class stellar_sdk.xdr.manage_data_op.ManageDataOp(data_name, data_value)`
XDR Source Code:

```
struct ManageDataOp
{
    string64 dataName;
    DataValue* dataValue; // set to null to clear
};
```

ManageDataResult

`class stellar_sdk.xdr.manage_data_result.ManageDataResult(code)`
XDR Source Code:

```
union ManageDataResult switch (ManageDataResultCode code)
{
    case MANAGE_DATA_SUCCESS:
        void;
    default:
        void;
};
```

ManageDataResultCode

class stellar_sdk.xdr.manage_data_result_code.**ManageDataResultCode**(value)

XDR Source Code:

```

enum ManageDataResultCode
{
    // codes considered as "success" for the operation
    MANAGE_DATA_SUCCESS = 0,
    // codes considered as "failure" for the operation
    MANAGE_DATA_NOT_SUPPORTED_YET =
        -1, // The network hasn't moved to this protocol change yet
    MANAGE_DATA_NAME_NOT_FOUND =
        -2, // Trying to remove a Data Entry that isn't there
    MANAGE_DATA_LOW_RESERVE = -3, // not enough funds to create a new Data Entry
    MANAGE_DATA_INVALID_NAME = -4 // Name not a valid string
};

```

ManageOfferEffect

class stellar_sdk.xdr.manage_offer_effect.**ManageOfferEffect**(value)

XDR Source Code:

```

enum ManageOfferEffect
{
    MANAGE_OFFER_CREATED = 0,
    MANAGE_OFFER_UPDATED = 1,
    MANAGE_OFFER_DELETED = 2
};

```

ManageOfferSuccessResult

class stellar_sdk.xdr.manage_offer_success_result.**ManageOfferSuccessResult**(offers_claimed, offer)

XDR Source Code:

```

struct ManageOfferSuccessResult
{
    // offers that got claimed while creating this offer
    ClaimAtom offersClaimed<>;

    union switch (ManageOfferEffect effect)
    {
        case MANAGE_OFFER_CREATED:
        case MANAGE_OFFER_UPDATED:
            OfferEntry offer;
        default:
            void;
    }
    offer;
};

```

ManageOfferSuccessResultOffer

`class stellar_sdk.xdr.manage_offer_success_result_offer.ManageOfferSuccessResultOffer`(*effect*,
*offer=**None*)

XDR Source Code:

```
union switch (ManageOfferEffect effect)
{
  case MANAGE_OFFER_CREATED:
  case MANAGE_OFFER_UPDATED:
    OfferEntry offer;
  default:
    void;
}
```

ManageSellOfferOp

`class stellar_sdk.xdr.manage_sell_offer_op.ManageSellOfferOp`(*selling*, *buying*, *amount*, *price*,
offer_id)

XDR Source Code:

```
struct ManageSellOfferOp
{
  Asset selling;
  Asset buying;
  int64 amount; // amount being sold. if set to 0, delete the offer
  Price price; // price of thing being sold in terms of what you are buying

  // 0=create a new offer, otherwise edit an existing offer
  int64 offerID;
};
```

ManageSellOfferResult

`class stellar_sdk.xdr.manage_sell_offer_result.ManageSellOfferResult`(*code*, *success=**None*)

XDR Source Code:

```
union ManageSellOfferResult switch (ManageSellOfferResultCode code)
{
  case MANAGE_SELL_OFFER_SUCCESS:
    ManageOfferSuccessResult success;
  default:
    void;
};
```

ManageSellOfferResultCode

`class stellar_sdk.xdr.manage_sell_offer_result_code.ManageSellOfferResultCode(value)`
 XDR Source Code:

```
enum ManageSellOfferResultCode
{
    // codes considered as "success" for the operation
    MANAGE_SELL_OFFER_SUCCESS = 0,

    // codes considered as "failure" for the operation
    MANAGE_SELL_OFFER_MALFORMED = -1, // generated offer would be invalid
    MANAGE_SELL_OFFER_SELL_NO_TRUST =
        -2, // no trust line for what we're selling
    MANAGE_SELL_OFFER_BUY_NO_TRUST = -3, // no trust line for what we're buying
    MANAGE_SELL_OFFER_SELL_NOT_AUTHORIZED = -4, // not authorized to sell
    MANAGE_SELL_OFFER_BUY_NOT_AUTHORIZED = -5, // not authorized to buy
    MANAGE_SELL_OFFER_LINE_FULL = -6, // can't receive more of what it's buying
    MANAGE_SELL_OFFER_UNDERFUNDED = -7, // doesn't hold what it's trying to sell
    MANAGE_SELL_OFFER_CROSS_SELF =
        -8, // would cross an offer from the same user
    MANAGE_SELL_OFFER_SELL_NO_ISSUER = -9, // no issuer for what we're selling
    MANAGE_SELL_OFFER_BUY_NO_ISSUER = -10, // no issuer for what we're buying

    // update errors
    MANAGE_SELL_OFFER_NOT_FOUND =
        -11, // offerID does not match an existing offer

    MANAGE_SELL_OFFER_LOW_RESERVE =
        -12 // not enough funds to create a new Offer
};
```

Memo

`class stellar_sdk.xdr.memo.Memo(type, text=None, id=None, hash=None, ret_hash=None)`
 XDR Source Code:

```
union Memo switch (MemoType type)
{
    case MEMO_NONE:
        void;
    case MEMO_TEXT:
        string text<28>;
    case MEMO_ID:
        uint64 id;
    case MEMO_HASH:
        Hash hash; // the hash of what to pull from the content server
    case MEMO_RETURN:
        Hash retHash; // the hash of the tx you are rejecting
};
```

MemoType

class stellar_sdk.xdr.memo_type.MemoType(*value*)

XDR Source Code:

```
enum MemoType
{
    MEMO_NONE = 0,
    MEMO_TEXT = 1,
    MEMO_ID = 2,
    MEMO_HASH = 3,
    MEMO_RETURN = 4
};
```

MessageType

class stellar_sdk.xdr.message_type.MessageType(*value*)

XDR Source Code:

```
enum MessageType
{
    ERROR_MSG = 0,
    AUTH = 2,
    DONT_HAVE = 3,

    GET_PEERS = 4, // gets a list of peers this guy knows about
    PEERS = 5,

    GET_TX_SET = 6, // gets a particular txset by hash
    TX_SET = 7,

    TRANSACTION = 8, // pass on a tx you have heard about

    // SCP
    GET_SCP_QUORUMSET = 9,
    SCP_QUORUMSET = 10,
    SCP_MESSAGE = 11,
    GET_SCP_STATE = 12,

    // new messages
    HELLO = 13,

    SURVEY_REQUEST = 14,
    SURVEY_RESPONSE = 15
};
```

MuxedAccount

class stellar_sdk.xdr.muxed_account.**MuxedAccount**(*type*, *ed25519=None*, *med25519=None*)
XDR Source Code:

```
union MuxedAccount switch (CryptoKeyType type)
{
  case KEY_TYPE_ED25519:
    uint256 ed25519;
  case KEY_TYPE_MUXED_ED25519:
    struct
    {
      uint64 id;
      uint256 ed25519;
    } med25519;
};
```

MuxedAccountMed25519

class stellar_sdk.xdr.muxed_account_med25519.**MuxedAccountMed25519**(*id*, *ed25519*)
XDR Source Code:

```
struct
{
  uint64 id;
  uint256 ed25519;
}
```

NodeID

class stellar_sdk.xdr.node_id.**NodeID**(*node_id*)
XDR Source Code:

```
typedef PublicKey NodeID;
```

OfferEntry

class stellar_sdk.xdr.offer_entry.**OfferEntry**(*seller_id*, *offer_id*, *selling*, *buying*, *amount*, *price*, *flags*, *ext*)

XDR Source Code:

```
struct OfferEntry
{
  AccountID sellerID;
  int64 offerID;
  Asset selling; // A
  Asset buying; // B
  int64 amount; // amount of A

  /* price for this offer:
```

(continues on next page)

(continued from previous page)

```

    price of A in terms of B
    price=AmountB/AmountA=priceNumerator/priceDenominator
    price is after fees
    */
    Price price;
    uint32 flags; // see OfferEntryFlags

    // reserved for future use
    union switch (int v)
    {
    case 0:
        void;
    }
    ext;
};

```

OfferEntryExt

class stellar_sdk.xdr.offer_entry_ext.OfferEntryExt(*v*)

XDR Source Code:

```

union switch (int v)
{
case 0:
    void;
}

```

OfferEntryFlags

class stellar_sdk.xdr.offer_entry_flags.OfferEntryFlags(*value*)

XDR Source Code:

```

enum OfferEntryFlags
{
    // issuer has authorized account to perform transactions with its credit
    PASSIVE_FLAG = 1
};

```

Opaque

class stellar_sdk.xdr.base.Opaque(*value, size, fixed*)

Operation

`class stellar_sdk.xdr.operation.Operation(source_account, body)`

XDR Source Code:

```

struct Operation
{
    // sourceAccount is the account used to run the operation
    // if not set, the runtime defaults to "sourceAccount" specified at
    // the transaction level
    MuxedAccount* sourceAccount;

    union switch (OperationType type)
    {
        case CREATE_ACCOUNT:
            CreateAccountOp createAccountOp;
        case PAYMENT:
            PaymentOp paymentOp;
        case PATH_PAYMENT_STRICT_RECEIVE:
            PathPaymentStrictReceiveOp pathPaymentStrictReceiveOp;
        case MANAGE_SELL_OFFER:
            ManageSellOfferOp manageSellOfferOp;
        case CREATE_PASSIVE_SELL_OFFER:
            CreatePassiveSellOfferOp createPassiveSellOfferOp;
        case SET_OPTIONS:
            SetOptionsOp setOptionsOp;
        case CHANGE_TRUST:
            ChangeTrustOp changeTrustOp;
        case ALLOW_TRUST:
            AllowTrustOp allowTrustOp;
        case ACCOUNT_MERGE:
            MuxedAccount destination;
        case INFLATION:
            void;
        case MANAGE_DATA:
            ManageDataOp manageDataOp;
        case BUMP_SEQUENCE:
            BumpSequenceOp bumpSequenceOp;
        case MANAGE_BUY_OFFER:
            ManageBuyOfferOp manageBuyOfferOp;
        case PATH_PAYMENT_STRICT_SEND:
            PathPaymentStrictSendOp pathPaymentStrictSendOp;
        case CREATE_CLAIMABLE_BALANCE:
            CreateClaimableBalanceOp createClaimableBalanceOp;
        case CLAIM_CLAIMABLE_BALANCE:
            ClaimClaimableBalanceOp claimClaimableBalanceOp;
        case BEGIN_SPONSORING_FUTURE_RESERVES:
            BeginSponsoringFutureReservesOp beginSponsoringFutureReservesOp;
        case END_SPONSORING_FUTURE_RESERVES:
            void;
        case REVOKE_SPONSORSHIP:
            RevokeSponsorshipOp revokeSponsorshipOp;
        case CLAWBACK:

```

(continues on next page)

(continued from previous page)

```

        ClawbackOp clawbackOp;
    case CLAWBACK_CLAIMABLE_BALANCE:
        ClawbackClaimableBalanceOp clawbackClaimableBalanceOp;
    case SET_TRUST_LINE_FLAGS:
        SetTrustLineFlagsOp setTrustLineFlagsOp;
    case LIQUIDITY_POOL_DEPOSIT:
        LiquidityPoolDepositOp liquidityPoolDepositOp;
    case LIQUIDITY_POOL_WITHDRAW:
        LiquidityPoolWithdrawOp liquidityPoolWithdrawOp;
    }
    body;
};

```

OperationBody

```

class stellar_sdk.xdr.operation_body.OperationBody(type, create_account_op=None,
                                                    payment_op=None,
                                                    path_payment_strict_receive_op=None,
                                                    manage_sell_offer_op=None,
                                                    create_passive_sell_offer_op=None,
                                                    set_options_op=None, change_trust_op=None,
                                                    allow_trust_op=None, destination=None,
                                                    manage_data_op=None,
                                                    bump_sequence_op=None,
                                                    manage_buy_offer_op=None,
                                                    path_payment_strict_send_op=None,
                                                    create_claimable_balance_op=None,
                                                    claim_claimable_balance_op=None,
                                                    begin_sponsoring_future_reserves_op=None,
                                                    revoke_sponsorship_op=None,
                                                    clawback_op=None,
                                                    clawback_claimable_balance_op=None,
                                                    set_trust_line_flags_op=None,
                                                    liquidity_pool_deposit_op=None,
                                                    liquidity_pool_withdraw_op=None)

```

XDR Source Code:

```

union switch (OperationType type)
{
    case CREATE_ACCOUNT:
        CreateAccountOp createAccountOp;
    case PAYMENT:
        PaymentOp paymentOp;
    case PATH_PAYMENT_STRICT_RECEIVE:
        PathPaymentStrictReceiveOp pathPaymentStrictReceiveOp;
    case MANAGE_SELL_OFFER:
        ManageSellOfferOp manageSellOfferOp;
    case CREATE_PASSIVE_SELL_OFFER:
        CreatePassiveSellOfferOp createPassiveSellOfferOp;
    case SET_OPTIONS:
        SetOptionsOp setOptionsOp;
}

```

(continues on next page)

(continued from previous page)

```

case CHANGE_TRUST:
    ChangeTrustOp changeTrustOp;
case ALLOW_TRUST:
    AllowTrustOp allowTrustOp;
case ACCOUNT_MERGE:
    MuxedAccount destination;
case INFLATION:
    void;
case MANAGE_DATA:
    ManageDataOp manageDataOp;
case BUMP_SEQUENCE:
    BumpSequenceOp bumpSequenceOp;
case MANAGE_BUY_OFFER:
    ManageBuyOfferOp manageBuyOfferOp;
case PATH_PAYMENT_STRICT_SEND:
    PathPaymentStrictSendOp pathPaymentStrictSendOp;
case CREATE_CLAIMABLE_BALANCE:
    CreateClaimableBalanceOp createClaimableBalanceOp;
case CLAIM_CLAIMABLE_BALANCE:
    ClaimClaimableBalanceOp claimClaimableBalanceOp;
case BEGIN_SPONSORING_FUTURE_RESERVES:
    BeginSponsoringFutureReservesOp beginSponsoringFutureReservesOp;
case END_SPONSORING_FUTURE_RESERVES:
    void;
case REVOKE_SPONSORSHIP:
    RevokeSponsorshipOp revokeSponsorshipOp;
case CLAWBACK:
    ClawbackOp clawbackOp;
case CLAWBACK_CLAIMABLE_BALANCE:
    ClawbackClaimableBalanceOp clawbackClaimableBalanceOp;
case SET_TRUST_LINE_FLAGS:
    SetTrustLineFlagsOp setTrustLineFlagsOp;
case LIQUIDITY_POOL_DEPOSIT:
    LiquidityPoolDepositOp liquidityPoolDepositOp;
case LIQUIDITY_POOL_WITHDRAW:
    LiquidityPoolWithdrawOp liquidityPoolWithdrawOp;
}

```

OperationID

class stellar_sdk.xdr.operation_id.OperationID(*type, id=None*)

XDR Source Code:

```

union OperationID switch (EnvelopeType type)
{
case ENVELOPE_TYPE_OP_ID:
    struct
    {
        AccountID sourceAccount;
        SequenceNumber seqNum;
        uint32 opNum;
    }
}

```

(continues on next page)

(continued from previous page)

```

    } id;
};

```

OperationIDId

class stellar_sdk.xdr.operation_id_id.**OperationIDId**(*source_account, seq_num, op_num*)
XDR Source Code:

```

struct
{
    AccountID sourceAccount;
    SequenceNumber seqNum;
    uint32 opNum;
}

```

OperationMeta

class stellar_sdk.xdr.operation_meta.**OperationMeta**(*changes*)
XDR Source Code:

```

struct OperationMeta
{
    LedgerEntryChanges changes;
};

```

OperationResult

class stellar_sdk.xdr.operation_result.**OperationResult**(*code, tr=None*)
XDR Source Code:

```

union OperationResult switch (OperationResultCode code)
{
    case opINNER:
        union switch (OperationType type)
        {
            case CREATE_ACCOUNT:
                CreateAccountResult createAccountResult;
            case PAYMENT:
                PaymentResult paymentResult;
            case PATH_PAYMENT_STRICT_RECEIVE:
                PathPaymentStrictReceiveResult pathPaymentStrictReceiveResult;
            case MANAGE_SELL_OFFER:
                ManageSellOfferResult manageSellOfferResult;
            case CREATE_PASSIVE_SELL_OFFER:
                ManageSellOfferResult createPassiveSellOfferResult;
            case SET_OPTIONS:
                SetOptionsResult setOptionsResult;
            case CHANGE_TRUST:
                ChangeTrustResult changeTrustResult;

```

(continues on next page)

(continued from previous page)

```

case ALLOW_TRUST:
    AllowTrustResult allowTrustResult;
case ACCOUNT_MERGE:
    AccountMergeResult accountMergeResult;
case INFLATION:
    InflationResult inflationResult;
case MANAGE_DATA:
    ManageDataResult manageDataResult;
case BUMP_SEQUENCE:
    BumpSequenceResult bumpSeqResult;
case MANAGE_BUY_OFFER:
    ManageBuyOfferResult manageBuyOfferResult;
case PATH_PAYMENT_STRICT_SEND:
    PathPaymentStrictSendResult pathPaymentStrictSendResult;
case CREATE_CLAIMABLE_BALANCE:
    CreateClaimableBalanceResult createClaimableBalanceResult;
case CLAIM_CLAIMABLE_BALANCE:
    ClaimClaimableBalanceResult claimClaimableBalanceResult;
case BEGIN_SPONSORING_FUTURE_RESERVES:
    BeginSponsoringFutureReservesResult beginSponsoringFutureReservesResult;
case END_SPONSORING_FUTURE_RESERVES:
    EndSponsoringFutureReservesResult endSponsoringFutureReservesResult;
case REVOKE_SPONSORSHIP:
    RevokeSponsorshipResult revokeSponsorshipResult;
case CLAWBACK:
    ClawbackResult clawbackResult;
case CLAWBACK_CLAIMABLE_BALANCE:
    ClawbackClaimableBalanceResult clawbackClaimableBalanceResult;
case SET_TRUST_LINE_FLAGS:
    SetTrustLineFlagsResult setTrustLineFlagsResult;
case LIQUIDITY_POOL_DEPOSIT:
    LiquidityPoolDepositResult liquidityPoolDepositResult;
case LIQUIDITY_POOL_WITHDRAW:
    LiquidityPoolWithdrawResult liquidityPoolWithdrawResult;
    }
    tr;
default:
    void;
};

```

OperationResultCode

class stellar_sdk.xdr.operation_result_code.**OperationResultCode**(*value*)

XDR Source Code:

```

enum OperationResultCode
{
    opINNER = 0, // inner object result is valid

    opBAD_AUTH = -1, // too few valid signatures / wrong network
    opNO_ACCOUNT = -2, // source account was not found
}

```

(continues on next page)

(continued from previous page)

```

opNOT_SUPPORTED = -3,          // operation not supported at this time
opTOO_MANY_SUBENTRIES = -4, // max number of subentries already reached
opEXCEEDED_WORK_LIMIT = -5, // operation did too much work
opTOO_MANY_SPONSORING = -6 // account is sponsoring too many entries
};

```

OperationResultTr

```

class stellar_sdk.xdr.operation_result_tr.OperationResultTr(
    type, create_account_result=None,
    payment_result=None,
    path_payment_strict_receive_result=None,
    manage_sell_offer_result=None,
    create_passive_sell_offer_result=None,
    set_options_result=None,
    change_trust_result=None,
    allow_trust_result=None,
    account_merge_result=None,
    inflation_result=None,
    manage_data_result=None,
    bump_seq_result=None,
    manage_buy_offer_result=None,
    path_payment_strict_send_result=None,
    create_claimable_balance_result=None,
    claim_claimable_balance_result=None,
    begin_sponsoring_future_reserves_result=None,
    end_sponsoring_future_reserves_result=None,
    revoke_sponsorship_result=None,
    clawback_result=None,
    clawback_claimable_balance_result=None,
    set_trust_line_flags_result=None,
    liquidity_pool_deposit_result=None,
    liquidity_pool_withdraw_result=None)

```

XDR Source Code:

```

union switch (OperationType type)
{
    case CREATE_ACCOUNT:
        CreateAccountResult createAccountResult;
    case PAYMENT:
        PaymentResult paymentResult;
    case PATH_PAYMENT_STRICT_RECEIVE:
        PathPaymentStrictReceiveResult pathPaymentStrictReceiveResult;
    case MANAGE_SELL_OFFER:
        ManageSellOfferResult manageSellOfferResult;
    case CREATE_PASSIVE_SELL_OFFER:
        ManageSellOfferResult createPassiveSellOfferResult;
    case SET_OPTIONS:
        SetOptionsResult setOptionsResult;
}

```

(continues on next page)

(continued from previous page)

```

case CHANGE_TRUST:
    ChangeTrustResult changeTrustResult;
case ALLOW_TRUST:
    AllowTrustResult allowTrustResult;
case ACCOUNT_MERGE:
    AccountMergeResult accountMergeResult;
case INFLATION:
    InflationResult inflationResult;
case MANAGE_DATA:
    ManageDataResult manageDataResult;
case BUMP_SEQUENCE:
    BumpSequenceResult bumpSeqResult;
case MANAGE_BUY_OFFER:
    ManageBuyOfferResult manageBuyOfferResult;
case PATH_PAYMENT_STRICT_SEND:
    PathPaymentStrictSendResult pathPaymentStrictSendResult;
case CREATE_CLAIMABLE_BALANCE:
    CreateClaimableBalanceResult createClaimableBalanceResult;
case CLAIM_CLAIMABLE_BALANCE:
    ClaimClaimableBalanceResult claimClaimableBalanceResult;
case BEGIN_SPONSORING_FUTURE_RESERVES:
    BeginSponsoringFutureReservesResult beginSponsoringFutureReservesResult;
case END_SPONSORING_FUTURE_RESERVES:
    EndSponsoringFutureReservesResult endSponsoringFutureReservesResult;
case REVOKE_SPONSORSHIP:
    RevokeSponsorshipResult revokeSponsorshipResult;
case CLAWBACK:
    ClawbackResult clawbackResult;
case CLAWBACK_CLAIMABLE_BALANCE:
    ClawbackClaimableBalanceResult clawbackClaimableBalanceResult;
case SET_TRUST_LINE_FLAGS:
    SetTrustLineFlagsResult setTrustLineFlagsResult;
case LIQUIDITY_POOL_DEPOSIT:
    LiquidityPoolDepositResult liquidityPoolDepositResult;
case LIQUIDITY_POOL_WITHDRAW:
    LiquidityPoolWithdrawResult liquidityPoolWithdrawResult;
}

```

OperationType

class stellar_sdk.xdr.operation_type.**OperationType**(*value*)

XDR Source Code:

```

enum OperationType
{
    CREATE_ACCOUNT = 0,
    PAYMENT = 1,
    PATH_PAYMENT_STRICT_RECEIVE = 2,
    MANAGE_SELL_OFFER = 3,
    CREATE_PASSIVE_SELL_OFFER = 4,
    SET_OPTIONS = 5,
}

```

(continues on next page)

(continued from previous page)

```

CHANGE_TRUST = 6,
ALLOW_TRUST = 7,
ACCOUNT_MERGE = 8,
INFLATION = 9,
MANAGE_DATA = 10,
BUMP_SEQUENCE = 11,
MANAGE_BUY_OFFER = 12,
PATH_PAYMENT_STRICT_SEND = 13,
CREATE_CLAIMABLE_BALANCE = 14,
CLAIM_CLAIMABLE_BALANCE = 15,
BEGIN_SPONSORING_FUTURE_RESERVES = 16,
END_SPONSORING_FUTURE_RESERVES = 17,
REVOKE_SPONSORSHIP = 18,
CLAWBACK = 19,
CLAWBACK_CLAIMABLE_BALANCE = 20,
SET_TRUST_LINE_FLAGS = 21,
LIQUIDITY_POOL_DEPOSIT = 22,
LIQUIDITY_POOL_WITHDRAW = 23
};

```

PathPaymentStrictReceiveOp

```

class stellar_sdk.xdr.path_payment_strict_receive_op.PathPaymentStrictReceiveOp(send_asset,
                                                                              send_max,
                                                                              destination,
                                                                              dest_asset,
                                                                              dest_amount,
                                                                              path)

```

XDR Source Code:

```

struct PathPaymentStrictReceiveOp
{
    Asset sendAsset; // asset we pay with
    int64 sendMax; // the maximum amount of sendAsset to
                  // send (excluding fees).
                  // The operation will fail if can't be met

    MuxedAccount destination; // recipient of the payment
    Asset destAsset; // what they end up with
    int64 destAmount; // amount they end up with

    Asset path<5>; // additional hops it must go through to get there
};

```


PathPaymentStrictReceiveResult

```
class stellar_sdk.xdr.path_payment_strict_receive_result.PathPaymentStrictReceiveResult(code,
                                                                                       success=None,
                                                                                       no_issuer=None)
```

XDR Source Code:

```
union PathPaymentStrictReceiveResult switch (
    PathPaymentStrictReceiveResultCode code)
{
case PATH_PAYMENT_STRICT_RECEIVE_SUCCESS:
    struct
    {
        ClaimAtom offers<>;
        SimplePaymentResult last;
    } success;
case PATH_PAYMENT_STRICT_RECEIVE_NO_ISSUER:
    Asset noIssuer; // the asset that caused the error
default:
    void;
};
```

PathPaymentStrictReceiveResultCode

```
class stellar_sdk.xdr.path_payment_strict_receive_result_code.PathPaymentStrictReceiveResultCode(value)
XDR Source Code:
```

```
enum PathPaymentStrictReceiveResultCode
{
    // codes considered as "success" for the operation
    PATH_PAYMENT_STRICT_RECEIVE_SUCCESS = 0, // success

    // codes considered as "failure" for the operation
    PATH_PAYMENT_STRICT_RECEIVE_MALFORMED = -1, // bad input
    PATH_PAYMENT_STRICT_RECEIVE_UNDERFUNDED =
        -2, // not enough funds in source account
    PATH_PAYMENT_STRICT_RECEIVE_SRC_NO_TRUST =
        -3, // no trust line on source account
    PATH_PAYMENT_STRICT_RECEIVE_SRC_NOT_AUTHORIZED =
        -4, // source not authorized to transfer
    PATH_PAYMENT_STRICT_RECEIVE_NO_DESTINATION =
        -5, // destination account does not exist
    PATH_PAYMENT_STRICT_RECEIVE_NO_TRUST =
        -6, // dest missing a trust line for asset
    PATH_PAYMENT_STRICT_RECEIVE_NOT_AUTHORIZED =
        -7, // dest not authorized to hold asset
    PATH_PAYMENT_STRICT_RECEIVE_LINE_FULL =
        -8, // dest would go above their limit
    PATH_PAYMENT_STRICT_RECEIVE_NO_ISSUER = -9, // missing issuer on one asset
    PATH_PAYMENT_STRICT_RECEIVE_TOO_FEW_OFFERS =
        -10, // not enough offers to satisfy path
};
```

(continues on next page)

(continued from previous page)

```

PATH_PAYMENT_STRICT_RECEIVE_OFFER_CROSS_SELF =
    -11, // would cross one of its own offers
PATH_PAYMENT_STRICT_RECEIVE_OVER_SENDFEE = -12 // could not satisfy sendmax
};

```

PathPaymentStrictReceiveResultSuccess

```
class stellar_sdk.xdr.path_payment_strict_receive_result_success.PathPaymentStrictReceiveResultSuccess(
```

XDR Source Code:

```

struct
{
    ClaimAtom offers<>;
    SimplePaymentResult last;
}

```

PathPaymentStrictSendOp

```
class stellar_sdk.xdr.path_payment_strict_send_op.PathPaymentStrictSendOp(send_asset,
                                                                    send_amount,
                                                                    destination,
                                                                    dest_asset,
                                                                    dest_min, path)
```

XDR Source Code:

```

struct PathPaymentStrictSendOp
{
    Asset sendAsset; // asset we pay with
    int64 sendAmount; // amount of sendAsset to send (excluding fees)

    MuxedAccount destination; // recipient of the payment
    Asset destAsset; // what they end up with
    int64 destMin; // the minimum amount of dest asset to
                  // be received
                  // The operation will fail if it can't be met

    Asset path<5>; // additional hops it must go through to get there
};

```

PathPaymentStrictSendResult

```
class stellar_sdk.xdr.path_payment_strict_send_result.PathPaymentStrictSendResult(code,
                                                                              success=None,
                                                                              no_issuer=None)
```

XDR Source Code:

```
union PathPaymentStrictSendResult switch (PathPaymentStrictSendResultCode code)
{
  case PATH_PAYMENT_STRICT_SEND_SUCCESS:
    struct
    {
      ClaimAtom offers<>;
      SimplePaymentResult last;
    } success;
  case PATH_PAYMENT_STRICT_SEND_NO_ISSUER:
    Asset noIssuer; // the asset that caused the error
  default:
    void;
};
```

PathPaymentStrictSendResultCode

```
class stellar_sdk.xdr.path_payment_strict_send_result_code.PathPaymentStrictSendResultCode(value)
```

XDR Source Code:

```
enum PathPaymentStrictSendResultCode
{
  // codes considered as "success" for the operation
  PATH_PAYMENT_STRICT_SEND_SUCCESS = 0, // success

  // codes considered as "failure" for the operation
  PATH_PAYMENT_STRICT_SEND_MALFORMED = -1, // bad input
  PATH_PAYMENT_STRICT_SEND_UNDERFUNDED =
    -2, // not enough funds in source account
  PATH_PAYMENT_STRICT_SEND_SRC_NO_TRUST =
    -3, // no trust line on source account
  PATH_PAYMENT_STRICT_SEND_SRC_NOT_AUTHORIZED =
    -4, // source not authorized to transfer
  PATH_PAYMENT_STRICT_SEND_NO_DESTINATION =
    -5, // destination account does not exist
  PATH_PAYMENT_STRICT_SEND_NO_TRUST =
    -6, // dest missing a trust line for asset
  PATH_PAYMENT_STRICT_SEND_NOT_AUTHORIZED =
    -7, // dest not authorized to hold asset
  PATH_PAYMENT_STRICT_SEND_LINE_FULL = -8, // dest would go above their limit
  PATH_PAYMENT_STRICT_SEND_NO_ISSUER = -9, // missing issuer on one asset
  PATH_PAYMENT_STRICT_SEND_TOO_FEW_OFFERS =
    -10, // not enough offers to satisfy path
  PATH_PAYMENT_STRICT_SEND_OFFER_CROSS_SELF =
    -11, // would cross one of its own offers
};
```

(continues on next page)

(continued from previous page)

```
PATH_PAYMENT_STRICT_SEND_UNDER_DESTMIN = -12 // could not satisfy destMin
};
```

PathPaymentStrictSendResultSuccess

`class stellar_sdk.xdr.path_payment_strict_send_result_success.PathPaymentStrictSendResultSuccess`(*offers*, *last*)

XDR Source Code:

```
struct
{
    ClaimAtom offers<>;
    SimplePaymentResult last;
}
```

PaymentOp

`class stellar_sdk.xdr.payment_op.PaymentOp`(*destination*, *asset*, *amount*)

XDR Source Code:

```
struct PaymentOp
{
    MuxedAccount destination; // recipient of the payment
    Asset asset;              // what they end up with
    int64 amount;             // amount they end up with
};
```

PaymentResult

`class stellar_sdk.xdr.payment_result.PaymentResult`(*code*)

XDR Source Code:

```
union PaymentResult switch (PaymentResultCode code)
{
case PAYMENT_SUCCESS:
    void;
default:
    void;
};
```

PaymentResultCode

class stellar_sdk.xdr.payment_result_code.PaymentResultCode(*value*)

XDR Source Code:

```

enum PaymentResultCode
{
    // codes considered as "success" for the operation
    PAYMENT_SUCCESS = 0, // payment successfully completed

    // codes considered as "failure" for the operation
    PAYMENT_MALFORMED = -1, // bad input
    PAYMENT_UNDERFUNDED = -2, // not enough funds in source account
    PAYMENT_SRC_NO_TRUST = -3, // no trust line on source account
    PAYMENT_SRC_NOT_AUTHORIZED = -4, // source not authorized to transfer
    PAYMENT_NO_DESTINATION = -5, // destination account does not exist
    PAYMENT_NO_TRUST = -6, // destination missing a trust line for asset
    PAYMENT_NOT_AUTHORIZED = -7, // destination not authorized to hold asset
    PAYMENT_LINE_FULL = -8, // destination would go above their limit
    PAYMENT_NO_ISSUER = -9 // missing issuer on asset
};

```

PeerAddress

class stellar_sdk.xdr.peer_address.PeerAddress(*ip*, *port*, *num_failures*)

XDR Source Code:

```

struct PeerAddress
{
    union switch (IPAddrType type)
    {
        case IPv4:
            opaque ipv4[4];
        case IPv6:
            opaque ipv6[16];
    }
    ip;
    uint32 port;
    uint32 numFailures;
};

```

PeerAddressIp

class stellar_sdk.xdr.peer_address_ip.PeerAddressIp(*type*, *ipv4=None*, *ipv6=None*)

XDR Source Code:

```

union switch (IPAddrType type)
{
    case IPv4:
        opaque ipv4[4];
    case IPv6:

```

(continues on next page)

(continued from previous page)

```
    opaque ipv6[16];  
}
```

PeerStatList

class stellar_sdk.xdr.peer_stat_list.**PeerStatList**(*peer_stat_list*)

XDR Source Code:

```
typedef PeerStats PeerStatList<25>;
```

PeerStats

class stellar_sdk.xdr.peer_stats.**PeerStats**(*id*, *version_str*, *messages_read*, *messages_written*, *bytes_read*, *bytes_written*, *seconds_connected*, *unique_flood_bytes_rcv*, *duplicate_flood_bytes_rcv*, *unique_fetch_bytes_rcv*, *duplicate_fetch_bytes_rcv*, *unique_flood_message_rcv*, *duplicate_flood_message_rcv*, *unique_fetch_message_rcv*, *duplicate_fetch_message_rcv*)

XDR Source Code:

```
struct PeerStats  
{  
    NodeID id;  
    string versionStr<100>;  
    uint64 messagesRead;  
    uint64 messagesWritten;  
    uint64 bytesRead;  
    uint64 bytesWritten;  
    uint64 secondsConnected;  
  
    uint64 uniqueFloodBytesRcv;  
    uint64 duplicateFloodBytesRcv;  
    uint64 uniqueFetchBytesRcv;  
    uint64 duplicateFetchBytesRcv;  
  
    uint64 uniqueFloodMessageRcv;  
    uint64 duplicateFloodMessageRcv;  
    uint64 uniqueFetchMessageRcv;  
    uint64 duplicateFetchMessageRcv;  
};
```

PoolID

class stellar_sdk.xdr.pool_id.**PoolID**(*pool_id*)
XDR Source Code:

```
typedef Hash PoolID;
```

Price

class stellar_sdk.xdr.price.**Price**(*n, d*)
XDR Source Code:

```
struct Price
{
    int32 n; // numerator
    int32 d; // denominator
};
```

PublicKey

class stellar_sdk.xdr.public_key.**PublicKey**(*type, ed25519=None*)
XDR Source Code:

```
union PublicKey switch (PublicKeyType type)
{
    case PUBLIC_KEY_TYPE_ED25519:
        uint256 ed25519;
};
```

PublicKeyType

class stellar_sdk.xdr.public_key_type.**PublicKeyType**(*value*)
XDR Source Code:

```
enum PublicKeyType
{
    PUBLIC_KEY_TYPE_ED25519 = KEY_TYPE_ED25519
};
```

RevokeSponsorshipOp

class stellar_sdk.xdr.revoke_sponsorship_op.**RevokeSponsorshipOp**(*type, ledger_key=None, signer=None*)

XDR Source Code:

```
union RevokeSponsorshipOp switch (RevokeSponsorshipType type)
{
    case REVOKE_SPONSORSHIP_LEDGER_ENTRY:
        LedgerKey ledgerKey;
```

(continues on next page)

(continued from previous page)

```

case REVOKE_SPONSORSHIP_SIGNER:
    struct
    {
        AccountID accountID;
        SignerKey signerKey;
    } signer;
};

```

RevokeSponsorshipOpSigner

```

class stellar_sdk.xdr.revoke_sponsorship_op_signer.RevokeSponsorshipOpSigner(account_id,
                                                                    signer_key)

```

XDR Source Code:

```

struct
{
    AccountID accountID;
    SignerKey signerKey;
}

```

RevokeSponsorshipResult

```

class stellar_sdk.xdr.revoke_sponsorship_result.RevokeSponsorshipResult(code)

```

XDR Source Code:

```

union RevokeSponsorshipResult switch (RevokeSponsorshipResultCode code)
{
case REVOKE_SPONSORSHIP_SUCCESS:
    void;
default:
    void;
};

```

RevokeSponsorshipResultCode

```

class stellar_sdk.xdr.revoke_sponsorship_result_code.RevokeSponsorshipResultCode(value)

```

XDR Source Code:

```

enum RevokeSponsorshipResultCode
{
    // codes considered as "success" for the operation
    REVOKE_SPONSORSHIP_SUCCESS = 0,

    // codes considered as "failure" for the operation
    REVOKE_SPONSORSHIP_DOES_NOT_EXIST = -1,
    REVOKE_SPONSORSHIP_NOT_SPONSOR = -2,
    REVOKE_SPONSORSHIP_LOW_RESERVE = -3,
    REVOKE_SPONSORSHIP_ONLY_TRANSFERABLE = -4,
}

```

(continues on next page)

(continued from previous page)

```

    REVOKE_SPONSORSHIP_MALFORMED = -5
};

```

RevokeSponsorshipType

class stellar_sdk.xdr.revoke_sponsorship_type.RevokeSponsorshipType(*value*)

XDR Source Code:

```

enum RevokeSponsorshipType
{
    REVOKE_SPONSORSHIP_LEDGER_ENTRY = 0,
    REVOKE_SPONSORSHIP_SIGNER = 1
};

```

SCPBallot

class stellar_sdk.xdr.scp_ballot.SCPBallot(*counter*, *value*)

XDR Source Code:

```

struct SCPBallot
{
    uint32 counter; // n
    Value value;    // x
};

```

SCPEnvelope

class stellar_sdk.xdr.scp_envelope.SCPEnvelope(*statement*, *signature*)

XDR Source Code:

```

struct SCPEnvelope
{
    SCPStatement statement;
    Signature signature;
};

```

SCPHistoryEntry

class stellar_sdk.xdr.scp_history_entry.SCPHistoryEntry(*v*, *v0=None*)

XDR Source Code:

```

union SCPHistoryEntry switch (int v)
{
    case 0:
        SCPHistoryEntryV0 v0;
};

```

SCPHistoryEntryV0

class stellar_sdk.xdr.scp_history_entry_v0.SCPHistoryEntryV0(*quorum_sets*, *ledger_messages*)
XDR Source Code:

```
struct SCPHistoryEntryV0
{
    SCPQuorumSet quorumSets<>; // additional quorum sets used by ledgerMessages
    LedgerSCPMessages ledgerMessages;
};
```

SCPNomination

class stellar_sdk.xdr.scp_nomination.SCPNomination(*quorum_set_hash*, *votes*, *accepted*)
XDR Source Code:

```
struct SCPNomination
{
    Hash quorumSetHash; // D
    Value votes<>;      // X
    Value accepted<>;   // Y
};
```

SCPQuorumSet

class stellar_sdk.xdr.scp_quorum_set.SCPQuorumSet(*threshold*, *validators*, *inner_sets*)
XDR Source Code:

```
struct SCPQuorumSet
{
    uint32 threshold;
    NodeID validators<>;
    SCPQuorumSet innerSets<>;
};
```

SCPStatement

class stellar_sdk.xdr.scp_statement.SCPStatement(*node_id*, *slot_index*, *pledges*)
XDR Source Code:

```
struct SCPStatement
{
    NodeID nodeID; // v
    uint64 slotIndex; // i

    union switch (SCPStatementType type)
    {
        case SCP_ST_PREPARE:
            struct
            {
```

(continues on next page)

(continued from previous page)

```

        Hash quorumSetHash; // D
        SCPBallot ballot; // b
        SCPBallot* prepared; // p
        SCPBallot* preparedPrime; // p'
        uint32 nC; // c.n
        uint32 nH; // h.n
    } prepare;
    case SCP_ST_CONFIRM:
    struct
    {
        SCPBallot ballot; // b
        uint32 nPrepared; // p.n
        uint32 nCommit; // c.n
        uint32 nH; // h.n
        Hash quorumSetHash; // D
    } confirm;
    case SCP_ST_EXTERNALIZE:
    struct
    {
        SCPBallot commit; // c
        uint32 nH; // h.n
        Hash commitQuorumSetHash; // D used before EXTERNALIZE
    } externalize;
    case SCP_ST_NOMINATE:
        SCPNomination nominate;
    }
    pledges;
};

```

SCPStatementConfirm

`class stellar_sdk.xdr.scp_statement_confirm.SCPStatementConfirm`(*ballot, n_prepared, n_commit, n_h, quorum_set_hash*)

XDR Source Code:

```

struct
{
    SCPBallot ballot; // b
    uint32 nPrepared; // p.n
    uint32 nCommit; // c.n
    uint32 nH; // h.n
    Hash quorumSetHash; // D
}

```

SCPStatementExternalize

`class stellar_sdk.xdr.scp_statement_externalize.SCPStatementExternalize(`*commit, n_h, commit_quorum_set_hash*`)`

XDR Source Code:

```
struct
{
    SCPBallot commit;           // c
    uint32 nH;                 // h.n
    Hash commitQuorumSetHash; // D used before EXTERNALIZE
}
```

SCPStatementPledges

`class stellar_sdk.xdr.scp_statement_pledges.SCPStatementPledges(`*type, prepare=None, confirm=None, externalize=None, nominate=None*`)`

XDR Source Code:

```
union switch (SCPStatementType type)
{
    case SCP_ST_PREPARE:
        struct
        {
            Hash quorumSetHash; // D
            SCPBallot ballot;    // b
            SCPBallot* prepared; // p
            SCPBallot* preparedPrime; // p'
            uint32 nC;          // c.n
            uint32 nH;          // h.n
        } prepare;
    case SCP_ST_CONFIRM:
        struct
        {
            SCPBallot ballot; // b
            uint32 nPrepared; // p.n
            uint32 nCommit;   // c.n
            uint32 nH;        // h.n
            Hash quorumSetHash; // D
        } confirm;
    case SCP_ST_EXTERNALIZE:
        struct
        {
            SCPBallot commit;           // c
            uint32 nH;                 // h.n
            Hash commitQuorumSetHash; // D used before EXTERNALIZE
        } externalize;
    case SCP_ST_NOMINATE:
        SCPNomination nominate;
}
```

SCPStatementPrepare

```
class stellar_sdk.xdr.scp_statement_prepare.SCPStatementPrepare(quorum_set_hash, ballot,
                                                             prepared, prepared_prime, n_c,
                                                             n_h)
```

XDR Source Code:

```
struct
{
    Hash quorumSetHash;      // D
    SCPBallot ballot;       // b
    SCPBallot* prepared;    // p
    SCPBallot* preparedPrime; // p'
    uint32 nC;              // c.n
    uint32 nH;              // h.n
}
```

SCPStatementType

```
class stellar_sdk.xdr.scp_statement_type.SCPStatementType(value)
```

XDR Source Code:

```
enum SCPStatementType
{
    SCP_ST_PREPARE = 0,
    SCP_ST_CONFIRM = 1,
    SCP_ST_EXTERNALIZE = 2,
    SCP_ST_NOMINATE = 3
};
```

SequenceNumber

```
class stellar_sdk.xdr.sequence_number.SequenceNumber(sequence_number)
```

XDR Source Code:

```
typedef int64 SequenceNumber;
```

SetOptionsOp

```
class stellar_sdk.xdr.set_options_op.SetOptionsOp(inflation_dest, clear_flags, set_flags,
                                                  master_weight, low_threshold, med_threshold,
                                                  high_threshold, home_domain, signer)
```

XDR Source Code:

```
struct SetOptionsOp
{
    AccountID* inflationDest; // sets the inflation destination

    uint32* clearFlags; // which flags to clear
    uint32* setFlags;   // which flags to set
}
```

(continues on next page)

(continued from previous page)

```

// account threshold manipulation
uint32* masterWeight; // weight of the master account
uint32* lowThreshold;
uint32* medThreshold;
uint32* highThreshold;

string32* homeDomain; // sets the home domain

// Add, update or remove a signer for the account
// signer is deleted if the weight is 0
Signer* signer;
};

```

SetOptionsResult

`class stellar_sdk.xdr.set_options_result.SetOptionsResult(code)`

XDR Source Code:

```

union SetOptionsResult switch (SetOptionsResultCode code)
{
case SET_OPTIONS_SUCCESS:
    void;
default:
    void;
};

```

SetOptionsResultCode

`class stellar_sdk.xdr.set_options_result_code.SetOptionsResultCode(value)`

XDR Source Code:

```

enum SetOptionsResultCode
{
    // codes considered as "success" for the operation
    SET_OPTIONS_SUCCESS = 0,
    // codes considered as "failure" for the operation
    SET_OPTIONS_LOW_RESERVE = -1, // not enough funds to add a signer
    SET_OPTIONS_TOO_MANY_SIGNERS = -2, // max number of signers already reached
    SET_OPTIONS_BAD_FLAGS = -3, // invalid combination of clear/set flags
    SET_OPTIONS_INVALID_INFLATION = -4, // inflation account does not exist
    SET_OPTIONS_CANT_CHANGE = -5, // can no longer change this option
    SET_OPTIONS_UNKNOWN_FLAG = -6, // can't set an unknown flag
    SET_OPTIONS_THRESHOLD_OUT_OF_RANGE = -7, // bad value for weight/threshold
    SET_OPTIONS_BAD_SIGNER = -8, // signer cannot be masterkey
    SET_OPTIONS_INVALID_HOME_DOMAIN = -9, // malformed home domain
    SET_OPTIONS_AUTH_REVOCABLE_REQUIRED =
        -10 // auth revocable is required for clawback
};

```

SetTrustLineFlagsOp

class stellar_sdk.xdr.set_trust_line_flags_op.**SetTrustLineFlagsOp**(*trustor, asset, clear_flags, set_flags*)

XDR Source Code:

```
struct SetTrustLineFlagsOp
{
    AccountID trustor;
    Asset asset;

    uint32 clearFlags; // which flags to clear
    uint32 setFlags;   // which flags to set
};
```

SetTrustLineFlagsResult

class stellar_sdk.xdr.set_trust_line_flags_result.**SetTrustLineFlagsResult**(*code*)

XDR Source Code:

```
union SetTrustLineFlagsResult switch (SetTrustLineFlagsResultCode code)
{
    case SET_TRUST_LINE_FLAGS_SUCCESS:
        void;
    default:
        void;
};
```

SetTrustLineFlagsResultCode

class stellar_sdk.xdr.set_trust_line_flags_result_code.**SetTrustLineFlagsResultCode**(*value*)

XDR Source Code:

```
enum SetTrustLineFlagsResultCode
{
    // codes considered as "success" for the operation
    SET_TRUST_LINE_FLAGS_SUCCESS = 0,

    // codes considered as "failure" for the operation
    SET_TRUST_LINE_FLAGS_MALFORMED = -1,
    SET_TRUST_LINE_FLAGS_NO_TRUST_LINE = -2,
    SET_TRUST_LINE_FLAGS_CANT_REVOKE = -3,
    SET_TRUST_LINE_FLAGS_INVALID_STATE = -4
};
```

Signature

class stellar_sdk.xdr.signature.**Signature**(*signature*)
XDR Source Code:

```
typedef opaque Signature<64>;
```

SignatureHint

class stellar_sdk.xdr.signature_hint.**SignatureHint**(*signature_hint*)
XDR Source Code:

```
typedef opaque SignatureHint[4];
```

SignedSurveyRequestMessage

class stellar_sdk.xdr.signed_survey_request_message.**SignedSurveyRequestMessage**(*request_signature*,
request)
XDR Source Code:

```
struct SignedSurveyRequestMessage  
{  
    Signature requestSignature;  
    SurveyRequestMessage request;  
};
```

SignedSurveyResponseMessage

class stellar_sdk.xdr.signed_survey_response_message.**SignedSurveyResponseMessage**(*response_signature*,
response)
XDR Source Code:

```
struct SignedSurveyResponseMessage  
{  
    Signature responseSignature;  
    SurveyResponseMessage response;  
};
```

Signer

class stellar_sdk.xdr.signer.**Signer**(*key*, *weight*)
XDR Source Code:

```
struct Signer  
{  
    SignerKey key;  
    uint32 weight; // really only need 1 byte  
};
```


SignerKey

class stellar_sdk.xdr.signer_key.**SignerKey**(*type*, *ed25519=None*, *pre_auth_tx=None*, *hash_x=None*)
XDR Source Code:

```
union SignerKey switch (SignerKeyType type)
{
case SIGNER_KEY_TYPE_ED25519:
    uint256 ed25519;
case SIGNER_KEY_TYPE_PRE_AUTH_TX:
    /* SHA-256 Hash of TransactionSignaturePayload structure */
    uint256 preAuthTx;
case SIGNER_KEY_TYPE_HASH_X:
    /* Hash of random 256 bit preimage X */
    uint256 hashX;
};
```

SignerKeyType

class stellar_sdk.xdr.signer_key_type.**SignerKeyType**(*value*)
XDR Source Code:

```
enum SignerKeyType
{
    SIGNER_KEY_TYPE_ED25519 = KEY_TYPE_ED25519,
    SIGNER_KEY_TYPE_PRE_AUTH_TX = KEY_TYPE_PRE_AUTH_TX,
    SIGNER_KEY_TYPE_HASH_X = KEY_TYPE_HASH_X
};
```

SimplePaymentResult

class stellar_sdk.xdr.simple_payment_result.**SimplePaymentResult**(*destination*, *asset*, *amount*)
XDR Source Code:

```
struct SimplePaymentResult
{
    AccountID destination;
    Asset asset;
    int64 amount;
};
```

SponsorshipDescriptor

class stellar_sdk.xdr.sponsorship_descriptor.**SponsorshipDescriptor**(*sponsorship_descriptor*)
XDR Source Code:

```
typedef AccountID* SponsorshipDescriptor;
```

StellarMessage

```
class stellar_sdk.xdr.stellar_message.StellarMessage(type, error=None, hello=None, auth=None, dont_have=None, peers=None, tx_set_hash=None, tx_set=None, transaction=None, signed_survey_request_message=None, signed_survey_response_message=None, q_set_hash=None, q_set=None, envelope=None, get_scp_ledger_seq=None)
```

XDR Source Code:

```
union StellarMessage switch (MessageType type)
{
  case ERROR_MSG:
    Error error;
  case HELLO:
    Hello hello;
  case AUTH:
    Auth auth;
  case DONT_HAVE:
    DontHave dontHave;
  case GET_PEERS:
    void;
  case PEERS:
    PeerAddress peers<100>;

  case GET_TX_SET:
    uint256 txSetHash;
  case TX_SET:
    TransactionSet txSet;

  case TRANSACTION:
    TransactionEnvelope transaction;

  case SURVEY_REQUEST:
    SignedSurveyRequestMessage signedSurveyRequestMessage;

  case SURVEY_RESPONSE:
    SignedSurveyResponseMessage signedSurveyResponseMessage;

  // SCP
  case GET_SCP_QUORUMSET:
    uint256 qSetHash;
  case SCP_QUORUMSET:
    SCPQuorumSet qSet;
  case SCP_MESSAGE:
    SCPEnvelope envelope;
  case GET_SCP_STATE:
    uint32 getSCPLedgerSeq; // ledger seq requested ; if 0, requests the latest
};
```

StellarValue

class stellar_sdk.xdr.stellar_value.StellarValue(tx_set_hash, close_time, upgrades, ext)
XDR Source Code:

```
struct StellarValue
{
    Hash txSetHash;          // transaction set to apply to previous ledger
    TimePoint closeTime;    // network close time

    // upgrades to apply to the previous ledger (usually empty)
    // this is a vector of encoded 'LedgerUpgrade' so that nodes can drop
    // unknown steps during consensus if needed.
    // see notes below on 'LedgerUpgrade' for more detail
    // max size is dictated by number of upgrade types (+ room for future)
    UpgradeType upgrades<6>;

    // reserved for future use
    union switch (StellarValueType v)
    {
        case STELLAR_VALUE_BASIC:
            void;
        case STELLAR_VALUE_SIGNED:
            LedgerCloseValueSignature lcValueSignature;
    }
    ext;
};
```

StellarValueExt

class stellar_sdk.xdr.stellar_value_ext.StellarValueExt(v, lc_value_signature=None)
XDR Source Code:

```
union switch (StellarValueType v)
{
    case STELLAR_VALUE_BASIC:
        void;
    case STELLAR_VALUE_SIGNED:
        LedgerCloseValueSignature lcValueSignature;
}
```

StellarValueType

class stellar_sdk.xdr.stellar_value_type.StellarValueType(value)
XDR Source Code:

```
enum StellarValueType
{
    STELLAR_VALUE_BASIC = 0,
    STELLAR_VALUE_SIGNED = 1
};
```

String

`class stellar_sdk.xdr.base.String(value, size)`

String32

`class stellar_sdk.xdr.string32.String32(string32)`

XDR Source Code:

```
typedef string string32<32>;
```

String64

`class stellar_sdk.xdr.string64.String64(string64)`

XDR Source Code:

```
typedef string string64<64>;
```

SurveyMessageCommandType

`class stellar_sdk.xdr.survey_message_command_type.SurveyMessageCommandType(value)`

XDR Source Code:

```
enum SurveyMessageCommandType
{
    SURVEY_TOPOLOGY = 0
};
```

SurveyRequestMessage

`class stellar_sdk.xdr.survey_request_message.SurveyRequestMessage(surveyor_peer_id, surveyed_peer_id, ledger_num, encryption_key, command_type)`

XDR Source Code:

```
struct SurveyRequestMessage
{
    NodeID surveyorPeerID;
    NodeID surveyedPeerID;
    uint32 ledgerNum;
    Curve25519Public encryptionKey;
    SurveyMessageCommandType commandType;
};
```

SurveyResponseBody

class stellar_sdk.xdr.survey_response_body.SurveyResponseBody(*type*,
topology_response_body=None)

XDR Source Code:

```
union SurveyResponseBody switch (SurveyMessageCommandType type)
{
case SURVEY_TOPOLOGY:
    TopologyResponseBody topologyResponseBody;
};
```

SurveyResponseMessage

class stellar_sdk.xdr.survey_response_message.SurveyResponseMessage(*surveyor_peer_id*,
surveyed_peer_id,
ledger_num,
command_type,
encrypted_body)

XDR Source Code:

```
struct SurveyResponseMessage
{
    NodeID surveyorPeerID;
    NodeID surveyedPeerID;
    uint32 ledgerNum;
    SurveyMessageCommandType commandType;
    EncryptedBody encryptedBody;
};
```

ThresholdIndexes

class stellar_sdk.xdr.threshold_indexes.ThresholdIndexes(*value*)

XDR Source Code:

```
enum ThresholdIndexes
{
    THRESHOLD_MASTER_WEIGHT = 0,
    THRESHOLD_LOW = 1,
    THRESHOLD_MED = 2,
    THRESHOLD_HIGH = 3
};
```

Thresholds

class stellar_sdk.xdr.thresholds.**Thresholds**(*thresholds*)
XDR Source Code:

```
typedef opaque Thresholds[4];
```

TimeBounds

class stellar_sdk.xdr.time_bounds.**TimeBounds**(*min_time*, *max_time*)
XDR Source Code:

```
struct TimeBounds
{
    TimePoint minTime;
    TimePoint maxTime; // 0 here means no maxTime
};
```

TimePoint

class stellar_sdk.xdr.time_point.**TimePoint**(*time_point*)
XDR Source Code:

```
typedef uint64 TimePoint;
```

TopologyResponseBody

class stellar_sdk.xdr.topology_response_body.**TopologyResponseBody**(*inbound_peers*,
outbound_peers,
total_inbound_peer_count,
total_outbound_peer_count)
XDR Source Code:

```
struct TopologyResponseBody
{
    PeerStatList inboundPeers;
    PeerStatList outboundPeers;

    uint32 totalInboundPeerCount;
    uint32 totalOutboundPeerCount;
};
```

Transaction

class stellar_sdk.xdr.transaction.**Transaction**(*source_account, fee, seq_num, time_bounds, memo, operations, ext*)

XDR Source Code:

```
struct Transaction
{
    // account used to run the transaction
    MuxedAccount sourceAccount;

    // the fee the sourceAccount will pay
    uint32 fee;

    // sequence number to consume in the account
    SequenceNumber seqNum;

    // validity range (inclusive) for the last ledger close time
    TimeBounds* timeBounds;

    Memo memo;

    Operation operations<MAX_OPS_PER_TX>;

    // reserved for future use
    union switch (int v)
    {
        case 0:
            void;
    }
    ext;
};
```

TransactionEnvelope

class stellar_sdk.xdr.transaction_envelope.**TransactionEnvelope**(*type, v0=None, v1=None, fee_bump=None*)

XDR Source Code:

```
union TransactionEnvelope switch (EnvelopeType type)
{
    case ENVELOPE_TYPE_TX_V0:
        TransactionV0Envelope v0;
    case ENVELOPE_TYPE_TX:
        TransactionV1Envelope v1;
    case ENVELOPE_TYPE_TX_FEE_BUMP:
        FeeBumpTransactionEnvelope feeBump;
};
```

TransactionExt

class stellar_sdk.xdr.transaction_ext.TransactionExt(*v*)
XDR Source Code:

```
union switch (int v)
{
    case 0:
        void;
}
```

TransactionHistoryEntry

class stellar_sdk.xdr.transaction_history_entry.TransactionHistoryEntry(*ledger_seq*, *tx_set*,
ext)

XDR Source Code:

```
struct TransactionHistoryEntry
{
    uint32 ledgerSeq;
    TransactionSet txSet;

    // reserved for future use
    union switch (int v)
    {
        case 0:
            void;
    }
    ext;
};
```

TransactionHistoryEntryExt

class stellar_sdk.xdr.transaction_history_entry_ext.TransactionHistoryEntryExt(*v*)
XDR Source Code:

```
union switch (int v)
{
    case 0:
        void;
}
```


TransactionHistoryResultEntry

class stellar_sdk.xdr.transaction_history_result_entry.TransactionHistoryResultEntry(*ledger_seq*,
tx_result_set,
ext)

XDR Source Code:

```
struct TransactionHistoryResultEntry
{
    uint32 ledgerSeq;
    TransactionResultSet txResultSet;

    // reserved for future use
    union switch (int v)
    {
        case 0:
            void;
    }
    ext;
};
```

TransactionHistoryResultEntryExt

class stellar_sdk.xdr.transaction_history_result_entry_ext.TransactionHistoryResultEntryExt(*v*)
XDR Source Code:

```
union switch (int v)
{
    case 0:
        void;
}
```

TransactionMeta

class stellar_sdk.xdr.transaction_meta.TransactionMeta(*v*, *operations=None*, *v1=None*, *v2=None*)
XDR Source Code:

```
union TransactionMeta switch (int v)
{
    case 0:
        OperationMeta operations<>;
    case 1:
        TransactionMetaV1 v1;
    case 2:
        TransactionMetaV2 v2;
};
```

TransactionMetaV1

`class stellar_sdk.xdr.transaction_meta_v1.TransactionMetaV1(tx_changes, operations)`
XDR Source Code:

```
struct TransactionMetaV1
{
    LedgerEntryChanges txChanges; // tx level changes if any
    OperationMeta operations<>; // meta for each operation
};
```

TransactionMetaV2

`class stellar_sdk.xdr.transaction_meta_v2.TransactionMetaV2(tx_changes_before, operations, tx_changes_after)`

XDR Source Code:

```
struct TransactionMetaV2
{
    LedgerEntryChanges txChangesBefore; // tx level changes before operations
                                        // are applied if any
    OperationMeta operations<>; // meta for each operation
    LedgerEntryChanges txChangesAfter; // tx level changes after operations are
                                        // applied if any
};
```

TransactionResult

`class stellar_sdk.xdr.transaction_result.TransactionResult(fee_charged, result, ext)`
XDR Source Code:

```
struct TransactionResult
{
    int64 feeCharged; // actual fee charged for the transaction

    union switch (TransactionResultCode code)
    {
        case txFEE_BUMP_INNER_SUCCESS:
        case txFEE_BUMP_INNER_FAILED:
            InnerTransactionResultPair innerResultPair;
        case txSUCCESS:
        case txFAILED:
            OperationResult results<>;
        default:
            void;
    }
    result;

    // reserved for future use
    union switch (int v)
    {
```

(continues on next page)

(continued from previous page)

```

    case 0:
        void;
    }
    ext;
};

```

TransactionResultCode

`class stellar_sdk.xdr.transaction_result_code.TransactionResultCode(value)`

XDR Source Code:

```

enum TransactionResultCode
{
    txFEE_BUMP_INNER_SUCCESS = 1, // fee bump inner transaction succeeded
    txSUCCESS = 0,                // all operations succeeded

    txFAILED = -1, // one of the operations failed (none were applied)

    txTOO_EARLY = -2, // ledger closeTime before minTime
    txTOO_LATE = -3,  // ledger closeTime after maxTime
    txMISSING_OPERATION = -4, // no operation was specified
    txBAD_SEQ = -5,       // sequence number does not match source account

    txBAD_AUTH = -6, // too few valid signatures / wrong network
    txINSUFFICIENT_BALANCE = -7, // fee would bring account below reserve
    txNO_ACCOUNT = -8, // source account not found
    txINSUFFICIENT_FEE = -9, // fee is too small
    txBAD_AUTH_EXTRA = -10, // unused signatures attached to transaction
    txINTERNAL_ERROR = -11, // an unknown error occurred

    txNOT_SUPPORTED = -12, // transaction type not supported
    txFEE_BUMP_INNER_FAILED = -13, // fee bump inner transaction failed
    txBAD_SPONSORSHIP = -14 // sponsorship not confirmed
};

```

TransactionResultExt

`class stellar_sdk.xdr.transaction_result_ext.TransactionResultExt(v)`

XDR Source Code:

```

union switch (int v)
{
    case 0:
        void;
}

```

TransactionResultMeta

class stellar_sdk.xdr.transaction_result_meta.**TransactionResultMeta**(*result, fee_processing, tx_apply_processing*)

XDR Source Code:

```
struct TransactionResultMeta
{
    TransactionResultPair result;
    LedgerEntryChanges feeProcessing;
    TransactionMeta txApplyProcessing;
};
```

TransactionResultPair

class stellar_sdk.xdr.transaction_result_pair.**TransactionResultPair**(*transaction_hash, result*)

XDR Source Code:

```
struct TransactionResultPair
{
    Hash transactionHash;
    TransactionResult result; // result for the transaction
};
```

TransactionResultResult

class stellar_sdk.xdr.transaction_result_result.**TransactionResultResult**(*code, inner_result_pair=None, results=None*)

XDR Source Code:

```
union switch (TransactionResultCode code)
{
    case txFEE_BUMP_INNER_SUCCESS:
    case txFEE_BUMP_INNER_FAILED:
        InnerTransactionResultPair innerResultPair;
    case txSUCCESS:
    case txFAILED:
        OperationResult results<>;
    default:
        void;
}
```

TransactionResultSet

class stellar_sdk.xdr.transaction_result_set.TransactionResultSet(*results*)

XDR Source Code:

```
struct TransactionResultSet
{
    TransactionResultPair results<>;
};
```

TransactionSet

class stellar_sdk.xdr.transaction_set.TransactionSet(*previous_ledger_hash*, *txs*)

XDR Source Code:

```
struct TransactionSet
{
    Hash previousLedgerHash;
    TransactionEnvelope txs<>;
};
```

TransactionSignaturePayload

class stellar_sdk.xdr.transaction_signature_payload.TransactionSignaturePayload(*network_id*,
tagged_transaction)

XDR Source Code:

```
struct TransactionSignaturePayload
{
    Hash networkId;
    union switch (EnvelopeType type)
    {
        // Backwards Compatibility: Use ENVELOPE_TYPE_TX to sign ENVELOPE_TYPE_TX_V0
        case ENVELOPE_TYPE_TX:
            Transaction tx;
        case ENVELOPE_TYPE_TX_FEE_BUMP:
            FeeBumpTransaction feeBump;
    }
    taggedTransaction;
};
```

TransactionSignaturePayloadTaggedTransaction

class stellar_sdk.xdr.transaction_signature_payload_tagged_transaction.TransactionSignaturePayloadTaggedTransaction

XDR Source Code:

```
union switch (EnvelopeType type)
{
```

(continues on next page)

(continued from previous page)

```
// Backwards Compatibility: Use ENVELOPE_TYPE_TX to sign ENVELOPE_TYPE_TX_V0
case ENVELOPE_TYPE_TX:
    Transaction tx;
case ENVELOPE_TYPE_TX_FEE_BUMP:
    FeeBumpTransaction feeBump;
}
```

TransactionV0

```
class stellar_sdk.xdr.transaction_v0.TransactionV0(source_account_ed25519, fee, seq_num,
time_bounds, memo, operations, ext)
```

XDR Source Code:

```
struct TransactionV0
{
    uint256 sourceAccountEd25519;
    uint32 fee;
    SequenceNumber seqNum;
    TimeBounds* timeBounds;
    Memo memo;
    Operation operations<MAX_OPS_PER_TX>;
    union switch (int v)
    {
        case 0:
            void;
    }
    ext;
};
```

TransactionV0Envelope

```
class stellar_sdk.xdr.transaction_v0_envelope.TransactionV0Envelope(tx, signatures)
```

XDR Source Code:

```
struct TransactionV0Envelope
{
    TransactionV0 tx;
    /* Each decorated signature is a signature over the SHA256 hash of
     * a TransactionSignaturePayload */
    DecoratedSignature signatures<20>;
};
```

TransactionV0Ext

class stellar_sdk.xdr.transaction_v0_ext.TransactionV0Ext(*v*)
 XDR Source Code:

```
union switch (int v)
{
  case 0:
    void;
}
```

TransactionV1Envelope

class stellar_sdk.xdr.transaction_v1_envelope.TransactionV1Envelope(*tx, signatures*)
 XDR Source Code:

```
struct TransactionV1Envelope
{
  Transaction tx;
  /* Each decorated signature is a signature over the SHA256 hash of
   * a TransactionSignaturePayload */
  DecoratedSignature signatures<20>;
};
```

TrustLineAsset

class stellar_sdk.xdr.trust_line_asset.TrustLineAsset(*type, alpha_num4=None, alpha_num12=None, liquidity_pool_id=None*)

XDR Source Code:

```
union TrustLineAsset switch (AssetType type)
{
  case ASSET_TYPE_NATIVE: // Not credit
    void;

  case ASSET_TYPE_CREDIT_ALPHANUM4:
    AlphaNum4 alphaNum4;

  case ASSET_TYPE_CREDIT_ALPHANUM12:
    AlphaNum12 alphaNum12;

  case ASSET_TYPE_POOL_SHARE:
    PoolID liquidityPoolID;

  // add other asset types here in the future
};
```

TrustLineEntry

class stellar_sdk.xdr.trust_line_entry.TrustLineEntry(*account_id, asset, balance, limit, flags, ext*)
XDR Source Code:

```
struct TrustLineEntry
{
    AccountID accountID; // account this trustline belongs to
    TrustLineAsset asset; // type of asset (with issuer)
    int64 balance; // how much of this asset the user has.
                    // Asset defines the unit for this;

    int64 limit; // balance cannot be above this
    uint32 flags; // see TrustLineFlags

    // reserved for future use
    union switch (int v)
    {
        case 0:
            void;
        case 1:
            struct
            {
                Liabilities liabilities;

                union switch (int v)
                {
                    case 0:
                        void;
                    case 2:
                        TrustLineEntryExtensionV2 v2;
                }
                ext;
            } v1;
    }
    ext;
};
```

TrustLineEntryExt

class stellar_sdk.xdr.trust_line_entry_ext.TrustLineEntryExt(*v, v1=None*)
XDR Source Code:

```
union switch (int v)
{
    case 0:
        void;
    case 1:
        struct
        {
            Liabilities liabilities;
```

(continues on next page)

(continued from previous page)

```

        union switch (int v)
        {
            case 0:
                void;
            case 2:
                TrustLineEntryExtensionV2 v2;
        }
        ext;
    } v1;
}

```

TrustLineEntryExtensionV2

class stellar_sdk.xdr.trust_line_entry_extension_v2.**TrustLineEntryExtensionV2**(*liquidity_pool_use_count*, *ext*)

XDR Source Code:

```

struct TrustLineEntryExtensionV2
{
    int32 liquidityPoolUseCount;

    union switch (int v)
    {
        case 0:
            void;
    }
    ext;
};

```

TrustLineEntryExtensionV2Ext

class stellar_sdk.xdr.trust_line_entry_extension_v2_ext.**TrustLineEntryExtensionV2Ext**(*v*)

XDR Source Code:

```

union switch (int v)
{
    case 0:
        void;
}

```

TrustLineEntryV1

class stellar_sdk.xdr.trust_line_entry_v1.TrustLineEntryV1(*liabilities, ext*)

XDR Source Code:

```
struct
{
    Liabilities liabilities;

    union switch (int v)
    {
        case 0:
            void;
        case 2:
            TrustLineEntryExtensionV2 v2;
    }
    ext;
}
```

TrustLineEntryV1Ext

class stellar_sdk.xdr.trust_line_entry_v1_ext.TrustLineEntryV1Ext(*v, v2=None*)

XDR Source Code:

```
union switch (int v)
{
    case 0:
        void;
    case 2:
        TrustLineEntryExtensionV2 v2;
}
```

TrustLineFlags

class stellar_sdk.xdr.trust_line_flags.TrustLineFlags(*value*)

XDR Source Code:

```
enum TrustLineFlags
{
    // issuer has authorized account to perform transactions with its credit
    AUTHORIZED_FLAG = 1,
    // issuer has authorized account to maintain and reduce liabilities for its
    // credit
    AUTHORIZED_TO_MAINTAIN_LIABILITIES_FLAG = 2,
    // issuer has specified that it may clawback its credit, and that claimable
    // balances created with its credit may also be clawed back
    TRUSTLINE_CLAWBACK_ENABLED_FLAG = 4
};
```

Uint256

class stellar_sdk.xdr.uint256.**Uint256**(*uint256*)
 XDR Source Code:

```
typedef opaque uint256[32];
```

Uint32

class stellar_sdk.xdr.uint32.**Uint32**(*uint32*)
 XDR Source Code:

```
typedef unsigned int uint32;
```

Uint64

class stellar_sdk.xdr.uint64.**Uint64**(*uint64*)
 XDR Source Code:

```
typedef unsigned hyper uint64;
```

UnsignedHyper

class stellar_sdk.xdr.base.**UnsignedHyper**(*value*)

UnsignedInteger

class stellar_sdk.xdr.base.**UnsignedInteger**(*value*)

UpgradeEntryMeta

class stellar_sdk.xdr.upgrade_entry_meta.**UpgradeEntryMeta**(*upgrade*, *changes*)
 XDR Source Code:

```
struct UpgradeEntryMeta
{
    LedgerUpgrade upgrade;
    LedgerEntryChanges changes;
};
```

UpgradeType

class stellar_sdk.xdr.upgrade_type.**UpgradeType**(*upgrade_type*)
XDR Source Code:

```
typedef opaque UpgradeType<128>;
```

Value

class stellar_sdk.xdr.value.**Value**(*value*)
XDR Source Code:

```
typedef opaque Value<>;
```

Constants

stellar_sdk.xdr.constants.**LIQUIDITY_POOL_FEE_V18**: **int** = 30
const LIQUIDITY_POOL_FEE_V18 = 30;

stellar_sdk.xdr.constants.**MASK_ACCOUNT_FLAGS**: **int** = 7
const MASK_ACCOUNT_FLAGS = 0x7;

stellar_sdk.xdr.constants.**MASK_ACCOUNT_FLAGS_V17**: **int** = 15
const MASK_ACCOUNT_FLAGS_V17 = 0xF;

stellar_sdk.xdr.constants.**MASK_CLAIMABLE_BALANCE_FLAGS**: **int** = 1
const MASK_CLAIMABLE_BALANCE_FLAGS = 0x1;

stellar_sdk.xdr.constants.**MASK_OFFERENTRY_FLAGS**: **int** = 1
const MASK_OFFERENTRY_FLAGS = 1;

stellar_sdk.xdr.constants.**MASK_TRUSTLINE_FLAGS**: **int** = 1
const MASK_TRUSTLINE_FLAGS = 1;

stellar_sdk.xdr.constants.**MASK_TRUSTLINE_FLAGS_V13**: **int** = 3
const MASK_TRUSTLINE_FLAGS_V13 = 3;

stellar_sdk.xdr.constants.**MASK_TRUSTLINE_FLAGS_V17**: **int** = 7
const MASK_TRUSTLINE_FLAGS_V17 = 7;

stellar_sdk.xdr.constants.**MAX_OPS_PER_TX**: **int** = 100
const MAX_OPS_PER_TX = 100;

stellar_sdk.xdr.constants.**MAX_SIGNERS**: **int** = 20
const MAX_SIGNERS = 20;

STELLAR-MODEL

stellar-model allows you to parse the JSON returned by Stellar Horizon into the Python models, click [here](#) for more information.

LINKS

- Document: <https://stellar-sdk.readthedocs.io>
- Code: <https://github.com/StellarCN/py-stellar-base>
- Examples: <https://github.com/StellarCN/py-stellar-base/tree/v7/examples>
- Issue tracker: <https://github.com/StellarCN/py-stellar-base/issues>
- License: Apache License 2.0
- Releases: <https://pypi.org/project/stellar-sdk/>

THANKS

This document is based on [Stellar JavaScript SDK](#) documentation. Thank you to all the people who have already contributed to Stellar ecosystem!

GENINDEX

PYTHON MODULE INDEX

S

`stellar_sdk`, 21

A

- Account (class in stellar_sdk.account), 21
- account_id() (stellar_sdk.call_builder.call_builder_async.AccountsCallBuilder method), 53
- account_id() (stellar_sdk.call_builder.call_builder_sync.AccountsCallBuilder method), 24
- account_muxed (stellar_sdk.muxed_account.MuxedAccount property), 96
- AccountEntry (class in stellar_sdk.xdr.account_entry), 160
- AccountEntryExt (class in stellar_sdk.xdr.account_entry_ext), 160
- AccountEntryExtensionV1 (class in stellar_sdk.xdr.account_entry_extension_v1), 161
- AccountEntryExtensionV1Ext (class in stellar_sdk.xdr.account_entry_extension_v1_ext), 161
- AccountEntryExtensionV2 (class in stellar_sdk.xdr.account_entry_extension_v2), 161
- AccountEntryExtensionV2Ext (class in stellar_sdk.xdr.account_entry_extension_v2_ext), 162
- AccountFlags (class in stellar_sdk.xdr.account_flags), 162
- AccountID (class in stellar_sdk.xdr.account_id), 162
- AccountMerge (class in stellar_sdk.operation), 99
- AccountMergeResult (class in stellar_sdk.xdr.account_merge_result), 163
- AccountMergeResultCode (class in stellar_sdk.xdr.account_merge_result_code), 163
- AccountRequiresMemoError (class in stellar_sdk.sep.exceptions), 159
- accounts() (stellar_sdk.server.Server method), 116
- accounts() (stellar_sdk.server_async.ServerAsync method), 121
- AccountsCallBuilder (class in stellar_sdk.call_builder.call_builder_async), 53
- AccountsCallBuilder (class in stellar_sdk.call_builder.call_builder_sync), 24
- add_hash_memo() (stellar_sdk.transaction_builder.TransactionBuilder method), 136
- add_id_memo() (stellar_sdk.transaction_builder.TransactionBuilder method), 137
- add_memo() (stellar_sdk.transaction_builder.TransactionBuilder method), 137
- add_return_hash_memo() (stellar_sdk.transaction_builder.TransactionBuilder method), 137
- add_text_memo() (stellar_sdk.transaction_builder.TransactionBuilder method), 137
- add_time_bounds() (stellar_sdk.transaction_builder.TransactionBuilder method), 137
- AiohttpClient (class in stellar_sdk.client.aiohttp_client), 84
- AllowTrust (class in stellar_sdk.operation), 99
- AllowTrustOp (class in stellar_sdk.xdr.allow_trust_op), 163
- AllowTrustResult (class in stellar_sdk.xdr.allow_trust_result), 164
- AllowTrustResultCode (class in stellar_sdk.xdr.allow_trust_result_code), 164
- AlphaNum12 (class in stellar_sdk.xdr.alpha_num12), 164
- AlphaNum4 (class in stellar_sdk.xdr.alpha_num4), 165
- append_account_merge_op() (stellar_sdk.transaction_builder.TransactionBuilder method), 138
- append_allow_trust_op() (stellar_sdk.transaction_builder.TransactionBuilder method), 138
- append_begin_sponsoring_future_reserves_op() (stellar_sdk.transaction_builder.TransactionBuilder method), 138
- append_bump_sequence_op() (stellar_sdk.transaction_builder.TransactionBuilder method), 138
- append_change_trust_op() (stellar_sdk.transaction_builder.TransactionBuilder method), 138

- lar_sdk.transaction_builder.TransactionBuilder* method), 139
- `append_claim_claimable_balance_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 139
- `append_clawback_claimable_balance_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 139
- `append_clawback_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 139
- `append_create_account_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 139
- `append_create_claimable_balance_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 140
- `append_create_passive_sell_offer_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 140
- `append_ed25519_public_key_signer()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 140
- `append_end_sponsoring_future_reserves_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 140
- `append_hashx_signer()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 141
- `append_inflation_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 141
- `append_liquidity_pool_deposit_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 141
- `append_liquidity_pool_withdraw_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 141
- `append_manage_buy_offer_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 142
- `append_manage_data_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 142
- `append_manage_sell_offer_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 142
- `append_operation()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 143
- `append_path_payment_strict_receive_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 143
- `append_path_payment_strict_send_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 143
- `append_payment_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 143
- `append_pre_auth_tx_signer()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 144
- `append_revoke_account_sponsorship_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 144
- `append_revoke_claimable_balance_sponsorship_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 144
- `append_revoke_data_sponsorship_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 144
- `append_revoke_ed25519_public_key_signer_sponsorship_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 145
- `append_revoke_hashx_signer_sponsorship_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 145
- `append_revoke_liquidity_pool_sponsorship_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 145
- `append_revoke_offer_sponsorship_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 145
- `append_revoke_pre_auth_tx_signer_sponsorship_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 146
- `append_revoke_trustline_sponsorship_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 146
- `append_set_options_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 146
- `append_set_trust_line_flags_op()` (*stellar_sdk.transaction_builder.TransactionBuilder* method), 147
- `Asset` (class in *stellar_sdk.asset*), 22
- `Asset` (class in *stellar_sdk.xdr.asset*), 165
- `AssetCode` (class in *stellar_sdk.xdr.asset_code*), 165
- `AssetCode12` (class in *stellar_sdk.xdr.asset_code12*), 166
- `AssetCode4` (class in *stellar_sdk.xdr.asset_code4*), 166
- `AssetCodeInvalidError` (class in *stellar_sdk.exceptions*), 88
- `AssetIssuerInvalidError` (class in *stellar_sdk.exceptions*), 88
- `assets()` (*stellar_sdk.server.Server* method), 116
- `assets()` (*stellar_sdk.server_async.ServerAsync* method), 121

- AssetsCallBuilder (class in stellar_sdk.call_builder.call_builder_async), 55
- AssetsCallBuilder (class in stellar_sdk.call_builder.call_builder_sync), 26
- AssetType (class in stellar_sdk.xdr.asset_type), 166
- Auth (class in stellar_sdk.xdr.auth), 166
- AuthCert (class in stellar_sdk.xdr.auth_cert), 166
- AuthenticatedMessage (class in stellar_sdk.xdr.authenticated_message), 167
- AuthenticatedMessageV0 (class in stellar_sdk.xdr.authenticated_message_v0), 167
- AuthorizationFlag (class in stellar_sdk.operation.set_options), 109
- ## B
- BadFederationResponseError (class in stellar_sdk.sep.exceptions), 159
- BadRequestError (class in stellar_sdk.exceptions), 89
- BadResponseError (class in stellar_sdk.exceptions), 89
- BadSignatureError (class in stellar_sdk.exceptions), 87
- BaseAsyncClient (class in stellar_sdk.client.base_async_client), 82
- BaseHorizonError (class in stellar_sdk.exceptions), 89
- BaseRequestError (class in stellar_sdk.exceptions), 88
- BaseSyncClient (class in stellar_sdk.client.base_sync_client), 83
- BeginSponsoringFutureReserves (class in stellar_sdk.operation), 112
- BeginSponsoringFutureReservesOp (class in stellar_sdk.xdr.begin_sponsoring_future_reserves_op), 167
- BeginSponsoringFutureReservesResult (class in stellar_sdk.xdr.begin_sponsoring_future_reserves_result), 167
- BeginSponsoringFutureReservesResultCode (class in stellar_sdk.xdr.begin_sponsoring_future_reserves_result_code), 168
- Boolean (class in stellar_sdk.xdr.base), 168
- BucketEntry (class in stellar_sdk.xdr.bucket_entry), 168
- BucketEntryType (class in stellar_sdk.xdr.bucket_entry_type), 168
- BucketMetadata (class in stellar_sdk.xdr.bucket_metadata), 169
- BucketMetadataExt (class in stellar_sdk.xdr.bucket_metadata_ext), 169
- build() (stellar_sdk.transaction_builder.TransactionBuilder method), 147
- build_challenge_transaction() (in module stellar_sdk.sep.stellar_web_authentication), 154
- build_fee_bump_transaction() (stellar_sdk.transaction_builder.TransactionBuilder static method), 148
- BumpSequence (class in stellar_sdk.operation), 100
- BumpSequenceOp (class in stellar_sdk.xdr.bump_sequence_op), 169
- BumpSequenceResult (class in stellar_sdk.xdr.bump_sequence_result), 170
- BumpSequenceResultCode (class in stellar_sdk.xdr.bump_sequence_result_code), 170
- ## C
- call() (stellar_sdk.call_builder.call_builder_async.AccountsCallBuilder method), 53
- call() (stellar_sdk.call_builder.call_builder_async.AssetsCallBuilder method), 55
- call() (stellar_sdk.call_builder.call_builder_async.ClaimableBalancesCallBuilder method), 56
- call() (stellar_sdk.call_builder.call_builder_async.DataCallBuilder method), 58
- call() (stellar_sdk.call_builder.call_builder_async.EffectsCallBuilder method), 59
- call() (stellar_sdk.call_builder.call_builder_async.FeeStatsCallBuilder method), 61
- call() (stellar_sdk.call_builder.call_builder_async.LedgersCallBuilder method), 62
- call() (stellar_sdk.call_builder.call_builder_async.LiquidityPoolsCallBuilder method), 63
- call() (stellar_sdk.call_builder.call_builder_async.OffersCallBuilder method), 65
- call() (stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder method), 67
- call() (stellar_sdk.call_builder.call_builder_async.OrderbookCallBuilder method), 69
- call() (stellar_sdk.call_builder.call_builder_async.PaymentsCallBuilder method), 70
- call() (stellar_sdk.call_builder.call_builder_async.RootCallBuilder method), 72
- call() (stellar_sdk.call_builder.call_builder_async.StrictReceivePathsCallBuilder method), 74
- call() (stellar_sdk.call_builder.call_builder_async.StrictSendPathsCallBuilder method), 75
- call() (stellar_sdk.call_builder.call_builder_async.TradeAggregationsCallBuilder method), 77
- call() (stellar_sdk.call_builder.call_builder_async.TradesCallBuilder method), 78
- call() (stellar_sdk.call_builder.call_builder_async.TransactionsCallBuilder method), 80
- call() (stellar_sdk.call_builder.call_builder_sync.AccountsCallBuilder method), 24

call() (stellar_sdk.call_builder.call_builder_sync.AssetsCallBuilder method), 26
 call() (stellar_sdk.call_builder.call_builder_sync.ClaimableBalancesCallBuilder method), 27
 call() (stellar_sdk.call_builder.call_builder_sync.DataCallBuilder method), 29
 call() (stellar_sdk.call_builder.call_builder_sync.EffectsCallBuilder method), 30
 call() (stellar_sdk.call_builder.call_builder_sync.FeeStatsCallBuilder method), 32
 call() (stellar_sdk.call_builder.call_builder_sync.LedgersCallBuilder method), 33
 call() (stellar_sdk.call_builder.call_builder_sync.LiquidityClaimableBalanceEntry method), 34
 call() (stellar_sdk.call_builder.call_builder_sync.OffersCallBuilder method), 36
 call() (stellar_sdk.call_builder.call_builder_sync.OperationsCallBuilder method), 38
 call() (stellar_sdk.call_builder.call_builder_sync.OrderbookCallBuilder method), 40
 call() (stellar_sdk.call_builder.call_builder_sync.PaymentsCallBuilder method), 41
 call() (stellar_sdk.call_builder.call_builder_sync.RootCallBuilder method), 43
 call() (stellar_sdk.call_builder.call_builder_sync.StrictReserveCallBuilder method), 45
 call() (stellar_sdk.call_builder.call_builder_sync.StrictReserveCallBuilder method), 46
 call() (stellar_sdk.call_builder.call_builder_sync.TradeAggregationsCallBuilder method), 48
 call() (stellar_sdk.call_builder.call_builder_sync.TradesCallBuilder method), 49
 call() (stellar_sdk.call_builder.call_builder_sync.TransactionsCallBuilder method), 51
 can_sign() (stellar_sdk.keypair.Keypair method), 89
 ChallengeTransaction (class in stellar_sdk.sep.stellar_web_authentication), 158
 ChangeTrust (class in stellar_sdk.operation), 101
 ChangeTrustAsset (class in stellar_sdk.xdr.change_trust_asset), 170
 ChangeTrustOp (class in stellar_sdk.xdr.change_trust_op), 171
 ChangeTrustResult (class in stellar_sdk.xdr.change_trust_result), 171
 ChangeTrustResultCode (class in stellar_sdk.xdr.change_trust_result_code), 171
 check_if_asset_code_is_valid() (stellar_sdk.asset.Asset static method), 22
 CHINESE_SIMPLIFIED (stellar_sdk.sep.mnemonic.Language attribute), 152
 CHINESE_TRADITIONAL (stellar_sdk.sep.mnemonic.Language attribute), 152
 claimable_balance() (stellar_sdk.call_builder.call_builder_sync.ClaimableBalancesCallBuilder method), 56
 claimable_balances() (stellar_sdk.server.Server method), 117
 claimable_balances() (stellar_sdk.server_async.ServerAsync method), 121
 ClaimableBalanceEntry (class in stellar_sdk.xdr.claimable_balance_entry), 175
 ClaimableBalanceEntryExt (class in stellar_sdk.xdr.claimable_balance_entry_ext), 176
 ClaimableBalanceEntryExtensionV1 (class in stellar_sdk.xdr.claimable_balance_entry_extension_v1), 176
 ClaimableBalanceEntryExtensionV1Ext (class in stellar_sdk.xdr.claimable_balance_entry_extension_v1_ext), 176
 ClaimableBalanceFlags (class in stellar_sdk.xdr.claimable_balance_flags), 177
 ClaimableBalanceID (class in stellar_sdk.xdr.claimable_balance_id), 177
 ClaimableBalanceIDType (class in stellar_sdk.xdr.claimable_balance_id_type), 177
 ClaimableBalancesCallBuilder (class in stellar_sdk.call_builder.call_builder_sync), 56
 Claimant (class in stellar_sdk.operation), 109
 Claimant (class in stellar_sdk.xdr.claimant), 177
 ClaimantType (class in stellar_sdk.xdr.claimant_type), 178
 ClaimantV0 (class in stellar_sdk.xdr.claimant_v0), 178
 ClaimAtom (class in stellar_sdk.xdr.claim_atom), 172
 ClaimAtomType (class in stellar_sdk.xdr.claim_atom_type), 172
 ClaimClaimableBalance (class in stellar_sdk.operation), 111
 ClaimClaimableBalanceOp (class in stellar_sdk.xdr.claim_claimable_balance_op), 172
 ClaimClaimableBalanceResult (class in stellar_sdk.xdr.claim_claimable_balance_result), 172
 ClaimClaimableBalanceResultCode (class in stellar_sdk.xdr.claim_claimable_balance_result_code), 172

- 173
- `ClaimLiquidityAtom` (class in `stellar_sdk.xdr.claim_liquidity_atom`), 173
- `ClaimOfferAtom` (class in `stellar_sdk.xdr.claim_offer_atom`), 173
- `ClaimOfferAtomV0` (class in `stellar_sdk.xdr.claim_offer_atom_v0`), 174
- `ClaimPredicate` (class in `stellar_sdk.operation`), 110
- `ClaimPredicate` (class in `stellar_sdk.xdr.claim_predicate`), 174
- `ClaimPredicateGroup` (class in `stellar_sdk.operation.create_claimable_balance`), 111
- `ClaimPredicateType` (class in `stellar_sdk.operation.create_claimable_balance`), 111
- `ClaimPredicateType` (class in `stellar_sdk.xdr.claim_predicate_type`), 175
- `Clawback` (class in `stellar_sdk.operation`), 114
- `ClawbackClaimableBalance` (class in `stellar_sdk.operation`), 114
- `ClawbackClaimableBalanceOp` (class in `stellar_sdk.xdr.clawback_claimable_balance_op`), 178
- `ClawbackClaimableBalanceResult` (class in `stellar_sdk.xdr.clawback_claimable_balance_result`), 178
- `ClawbackClaimableBalanceResultCode` (class in `stellar_sdk.xdr.clawback_claimable_balance_result_code`), 179
- `ClawbackOp` (class in `stellar_sdk.xdr.clawback_op`), 179
- `ClawbackResult` (class in `stellar_sdk.xdr.clawback_result`), 179
- `ClawbackResultCode` (class in `stellar_sdk.xdr.clawback_result_code`), 179
- `close()` (`stellar_sdk.client.aiohttp_client.AiohttpClient` method), 84
- `close()` (`stellar_sdk.client.requests_client.RequestsClient` method), 85
- `close()` (`stellar_sdk.server.Server` method), 117
- `close()` (`stellar_sdk.server_async.ServerAsync` method), 121
- `ConnectionError` (class in `stellar_sdk.exceptions`), 88
- `CreateAccount` (class in `stellar_sdk.operation`), 101
- `CreateAccountOp` (class in `stellar_sdk.xdr.create_account_op`), 180
- `CreateAccountResult` (class in `stellar_sdk.xdr.create_account_result`), 180
- `CreateAccountResultCode` (class in `stellar_sdk.xdr.create_account_result_code`), 180
- `CreateClaimableBalance` (class in `stellar_sdk.operation`), 109
- `CreateClaimableBalanceOp` (class in `stellar_sdk.xdr.create_claimable_balance_op`), 181
- `CreateClaimableBalanceResult` (class in `stellar_sdk.xdr.create_claimable_balance_result`), 181
- `CreateClaimableBalanceResultCode` (class in `stellar_sdk.xdr.create_claimable_balance_result_code`), 181
- `CreatePassiveSellOffer` (class in `stellar_sdk.operation`), 102
- `CreatePassiveSellOfferOp` (class in `stellar_sdk.xdr.create_passive_sell_offer_op`), 182
- `CryptoKeyType` (class in `stellar_sdk.xdr.crypto_key_type`), 182
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.AccountsCallBuilder` method), 53
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.AssetsCallBuilder` method), 55
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.ClaimableBalances` method), 57
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.DataCallBuilder` method), 58
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.EffectsCallBuilder` method), 59
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.FeeStatsCallBuilder` method), 61
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.LedgersCallBuilder` method), 62
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.LiquidityPoolsBuilder` method), 64
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.OffersCallBuilder` method), 65
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder` method), 67
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.OrderbookCallBuilder` method), 69
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.PaymentsCallBuilder` method), 71
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.RootCallBuilder` method), 72
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.StrictReceivePathsCallBuilder` method), 74
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.StrictSendPathsCallBuilder` method), 76
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.TradeAggregationsCallBuilder` method), 77
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.TradesCallBuilder` method), 78
- `cursor()` (`stellar_sdk.call_builder.call_builder_async.TransactionsCallBuilder` method), 80
- `cursor()` (`stellar_sdk.call_builder.call_builder_sync.AccountsCallBuilder` method), 24

[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.AssetCallBuilder* method), 26
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.ClaimableBalanceCallBuilder* method), 28
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.DataCallBuilder* method), 29
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.EffectsCallBuilder* method), 30
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.FeeStatsCallBuilder* method), 32
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.LedgersCallBuilder* method), 33
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.LiquidityPoolsCallBuilder* method), 35
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.OffersCallBuilder* method), 36
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.OperationsCallBuilder* method), 38
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.OrderbookCallBuilder* method), 40
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.PaymentsCallBuilder* method), 42
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.RootCallBuilder* method), 43
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.StrictReceivePathsCallBuilder* method), 45
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.StrictSendPathsCallBuilder* method), 47
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.TradeAggregationCallBuilder* method), 48
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.TradeEffectsCallBuilder* method), 49
[cursor\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.TransactionsCallBuilder* method), 51
[Curve25519Public](#) (class in *stellar_sdk.xdr.curve25519_public*), 182
[Curve25519Secret](#) (class in *stellar_sdk.xdr.curve25519_secret*), 182
D
[Data](#) (class in *stellar_sdk.operation.revoke_sponsorship*), 113
[data\(\)](#) (*stellar_sdk.server.Server* method), 117
[data\(\)](#) (*stellar_sdk.server_async.ServerAsync* method), 121
[DataCallBuilder](#) (class in *stellar_sdk.call_builder.call_builder_async*), 58
[DataCallBuilder](#) (class in *stellar_sdk.call_builder.call_builder_sync*), 29
[DataEntry](#) (class in *stellar_sdk.xdr.data_entry*), 183
[DataEntryExt](#) (class in *stellar_sdk.xdr.data_entry_ext*), 183
[DataBuilder](#) (class in *stellar_sdk.xdr.data_value*), 183
[decode_ed25519_public_key\(\)](#) (*stellar_sdk.strkey.StrKey* static method), 127
[decode_ed25519_secret_seed\(\)](#) (*stellar_sdk.strkey.StrKey* static method), 127
[decode_muxed_account\(\)](#) (*stellar_sdk.strkey.StrKey* static method), 128
[decode_pre_auth_tx\(\)](#) (*stellar_sdk.strkey.StrKey* static method), 128
[decode_sha256_hash\(\)](#) (*stellar_sdk.strkey.StrKey* static method), 128
[decorated_signature](#) (class in *stellar_sdk.xdr.decorated_signature*), 183
[DontHave](#) (class in *stellar_sdk.xdr.dont_have*), 184
E
[Ed25519PublicKey](#) (class in *stellar_sdk.signer.Signer*), 125
[Ed25519PublicKeyInvalidError](#) (class in *stellar_sdk.exceptions*), 87
[Ed25519SecretSeedInvalidError](#) (class in *stellar_sdk.exceptions*), 87
[effects\(\)](#) (*stellar_sdk.server.Server* method), 117
[effects\(\)](#) (*stellar_sdk.server_async.ServerAsync* method), 121
[EffectsCallBuilder](#) (class in *stellar_sdk.call_builder.call_builder_async*), 30
[encode_ed25519_public_key\(\)](#) (*stellar_sdk.strkey.StrKey* static method), 127
[encode_ed25519_secret_seed\(\)](#) (*stellar_sdk.strkey.StrKey* static method), 128
[encode_muxed_account\(\)](#) (*stellar_sdk.strkey.StrKey* static method), 128
[encode_pre_auth_tx\(\)](#) (*stellar_sdk.strkey.StrKey* static method), 128
[encode_sha256_hash\(\)](#) (*stellar_sdk.strkey.StrKey* static method), 128
[encoded_signer_key](#) (*stellar_sdk.signer_key.SignerKey* property), 126
[EncryptedBody](#) (class in *stellar_sdk.xdr.encrypted_body*), 184
[EndSponsoringFutureReserves](#) (class in *stellar_sdk.operation*), 112

EndSponsoringFutureReservesResult (class in stellar_sdk.xdr.end_sponsoring_future_reserves_result), 184

EndSponsoringFutureReservesResultCode (class in stellar_sdk.xdr.end_sponsoring_future_reserves_result_code), 184

ENGLISH (stellar_sdk.sep.mnemonic.Language attribute), 152

EnvelopeType (class in stellar_sdk.xdr.envelope_type), 185

Error (class in stellar_sdk.xdr.error), 185

ErrorCode (class in stellar_sdk.xdr.error_code), 185

F

FeatureNotEnabledError (class in stellar_sdk.exceptions), 89

FederationRecord (class in stellar_sdk.sep.federation), 152

FederationServerNotFoundError (class in stellar_sdk.sep.exceptions), 159

fee_stats() (stellar_sdk.server.Server method), 117

fee_stats() (stellar_sdk.server_async.ServerAsync method), 121

FeeBumpTransaction (class in stellar_sdk.fee_bump_transaction), 133

FeeBumpTransaction (class in stellar_sdk.xdr.fee_bump_transaction), 185

FeeBumpTransactionEnvelope (class in stellar_sdk.fee_bump_transaction_envelope), 134

FeeBumpTransactionEnvelope (class in stellar_sdk.xdr.fee_bump_transaction_envelope), 186

FeeBumpTransactionExt (class in stellar_sdk.xdr.fee_bump_transaction_ext), 186

FeeBumpTransactionInnerTx (class in stellar_sdk.xdr.fee_bump_transaction_inner_tx), 186

FeeStatsCallBuilder (class in stellar_sdk.call_builder.call_builder_async), 61

FeeStatsCallBuilder (class in stellar_sdk.call_builder.call_builder_sync), 32

fetch_base_fee() (stellar_sdk.server.Server method), 117

fetch_base_fee() (stellar_sdk.server_async.ServerAsync method), 121

fetch_stellar_toml() (in module stellar_sdk.sep.stellar_toml), 150

fetch_stellar_toml_async() (in module stellar_sdk.sep.stellar_toml), 150

for_account() (stellar_sdk.call_builder.call_builder_async.EffectsCallBuilder method), 59

for_account() (stellar_sdk.call_builder.call_builder_async.LiquidityPoolsCallBuilder method), 64

for_account() (stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder method), 67

for_account() (stellar_sdk.call_builder.call_builder_async.PaymentsCallBuilder method), 71

for_account() (stellar_sdk.call_builder.call_builder_async.TradesCallBuilder method), 78

for_account() (stellar_sdk.call_builder.call_builder_async.TransactionsCallBuilder method), 80

for_account() (stellar_sdk.call_builder.call_builder_sync.EffectsCallBuilder method), 30

for_account() (stellar_sdk.call_builder.call_builder_sync.LiquidityPoolsCallBuilder method), 35

for_account() (stellar_sdk.call_builder.call_builder_sync.OperationsCallBuilder method), 38

for_account() (stellar_sdk.call_builder.call_builder_sync.PaymentsCallBuilder method), 42

for_account() (stellar_sdk.call_builder.call_builder_sync.TradesCallBuilder method), 49

for_account() (stellar_sdk.call_builder.call_builder_sync.TransactionsCallBuilder method), 51

for_asset() (stellar_sdk.call_builder.call_builder_async.AccountsCallBuilder method), 53

for_asset() (stellar_sdk.call_builder.call_builder_async.ClaimableBalanceCallBuilder method), 57

for_asset() (stellar_sdk.call_builder.call_builder_sync.AccountsCallBuilder method), 24

for_asset() (stellar_sdk.call_builder.call_builder_sync.ClaimableBalanceCallBuilder method), 28

for_asset_pair() (stellar_sdk.call_builder.call_builder_async.TradesCallBuilder method), 78

for_asset_pair() (stellar_sdk.call_builder.call_builder_sync.TradesCallBuilder method), 49

for_buying() (stellar_sdk.call_builder.call_builder_async.OffersCallBuilder method), 65

for_buying() (stellar_sdk.call_builder.call_builder_sync.OffersCallBuilder method), 36

for_claimable_balance() (stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder method), 67

for_claimable_balance() (stellar_sdk.call_builder.call_builder_async.TransactionsCallBuilder method), 80

for_claimable_balance() (stellar_sdk.call_builder.call_builder_sync.OperationsCallBuilder method), 38

for_claimable_balance() (stellar_sdk.call_builder.call_builder_sync.TransactionsCallBuilder method), 51

method), 60
for_transaction() (*stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder* class method), 129
method), 68
for_transaction() (*stellar_sdk.call_builder.call_builder_async.PaymentsCallBuilder* class method), 133
method), 71
for_transaction() (*stellar_sdk.call_builder.call_builder_sync.EffectsCallBuilder* class method), 134
method), 31
for_transaction() (*stellar_sdk.call_builder.call_builder_sync.OperationsCallBuilder* class method), 92
method), 39
for_transaction() (*stellar_sdk.call_builder.call_builder_sync.PaymentsCallBuilder* class method), 93
method), 42
FRENCH (*stellar_sdk.sep.mnemonic.Language* attribute), 152
from_account() (*stellar_sdk.muxed_account.MuxedAccount* class method), 97
from_mnemonic_phrase() (*stellar_sdk.keypair.Keypair* class method), 89
from_public_key() (*stellar_sdk.keypair.Keypair* class method), 90
from_raw_ed25519_public_key() (*stellar_sdk.keypair.Keypair* class method), 90
from_raw_ed25519_seed() (*stellar_sdk.keypair.Keypair* class method), 90
from_raw_price() (*stellar_sdk.price.Price* class method), 116
from_secret() (*stellar_sdk.keypair.Keypair* class method), 90
from_txrep() (in module *stellar_sdk.sep.txrep*), 159
from_uri() (*stellar_sdk.sep.stellar_uri.PayStellarUri* class method), 153
from_uri() (*stellar_sdk.sep.stellar_uri.TransactionStellarUri* class method), 153
from_xdr() (*stellar_sdk.fee_bump_transaction.FeeBumpTransaction* class method), 133
from_xdr() (*stellar_sdk.fee_bump_transaction_envelope.FeeBumpTransactionEnvelope* class method), 134
from_xdr() (*stellar_sdk.transaction.Transaction* class method), 130
from_xdr() (*stellar_sdk.transaction_builder.TransactionBuilder* static method), 148
from_xdr() (*stellar_sdk.transaction_envelope.TransactionEnvelope* class method), 131
from_xdr_amount() (in module *stellar_sdk.xdr.utils*), 149
from_xdr_amount() (*stellar_sdk.operation.Operation* static method), 98
from_xdr_object() (*stellar_sdk.asset.Asset* class method), 22
from_xdr_object() (*stellar_sdk.decorated_signature.DecoratedSignature* class method), 129
from_xdr_object() (*stellar_sdk.fee_bump_transaction.FeeBumpTransaction* class method), 133
from_xdr_object() (*stellar_sdk.fee_bump_transaction_envelope.FeeBumpTransactionEnvelope* class method), 134
from_xdr_object() (*stellar_sdk.liquidity_pool_asset.LiquidityPoolAsset* class method), 92
from_xdr_object() (*stellar_sdk.liquidity_pool_id.LiquidityPoolId* class method), 93
from_xdr_object() (*stellar_sdk.memo.HashMemo* class method), 95
from_xdr_object() (*stellar_sdk.memo.IdMemo* class method), 95
from_xdr_object() (*stellar_sdk.memo.Memo* static method), 94
from_xdr_object() (*stellar_sdk.memo.NoneMemo* class method), 94
from_xdr_object() (*stellar_sdk.memo.ReturnHashMemo* class method), 96
from_xdr_object() (*stellar_sdk.memo.TextMemo* class method), 95
from_xdr_object() (*stellar_sdk.muxed_account.MuxedAccount* class method), 97
from_xdr_object() (*stellar_sdk.operation.AccountMerge* class method), 99
from_xdr_object() (*stellar_sdk.operation.AllowTrust* class method), 100
from_xdr_object() (*stellar_sdk.operation.BeginSponsoringFutureReserves* class method), 100
from_xdr_object() (*stellar_sdk.operation.FeeBumpSequence* class method), 100
from_xdr_object() (*stellar_sdk.operation.ChangeTrust* class method), 101
from_xdr_object() (*stellar_sdk.operation.ClaimClaimableBalance* class method), 111
from_xdr_object() (*stellar_sdk.operation.Clawback* class method), 114
from_xdr_object() (*stellar_sdk.operation.ClawbackClaimableBalance* class method), 114
from_xdr_object() (*stellar_sdk.operation.ClawbackClaimableBalance* class method), 114

<code>lar_sdk.operation.CreateAccount</code>	class	<code>method</code>), 129
<code>method</code>), 101		
<code>from_xdr_object()</code>	(stellar_sdk.operation.CreateClaimableBalance	<code>from_xdr_object()</code> (stellar_sdk.transaction.Transaction class method), 130
<code>class method</code>), 109		
<code>from_xdr_object()</code>	(stellar_sdk.operation.CreatePassiveSellOffer	<code>from_xdr_object()</code> (stellar_sdk.transaction_envelope.TransactionEnvelope class method), 131
<code>class method</code>), 102		
<code>from_xdr_object()</code>	(stellar_sdk.operation.EndSponsoringFutureReserves	G
<code>class method</code>), 112		<code>generate_mnemonic_phrase()</code> (stellar_sdk.keypair.Keypair static method), 91
<code>from_xdr_object()</code>	(stellar_sdk.operation.Inflation	<code>get()</code> (stellar_sdk.client.aiohttp_client.AiohttpClient method), 84
<code>class method</code>), 102		<code>get()</code> (stellar_sdk.client.base_async_client.BaseAsyncClient method), 82
<code>from_xdr_object()</code>	(stellar_sdk.operation.LiquidityPoolDeposit	<code>get()</code> (stellar_sdk.client.base_sync_client.BaseSyncClient method), 83
<code>class method</code>), 103		<code>get()</code> (stellar_sdk.client.requests_client.RequestsClient method), 85
<code>from_xdr_object()</code>	(stellar_sdk.operation.LiquidityPoolWithdraw	<code>get()</code> (stellar_sdk.client.simple_requests_client.SimpleRequestsClient method), 86
<code>class method</code>), 104		<code>get_claimable_balance_id()</code> (stellar_sdk.transaction.Transaction method), 131
<code>from_xdr_object()</code>	(stellar_sdk.operation.ManageBuyOffer	<code>get_source_from_xdr_obj()</code> (stellar_sdk.operation.Operation static method), 98
<code>class method</code>), 104		<code>guess_asset_type()</code> (stellar_sdk.asset.Asset method), 23
<code>from_xdr_object()</code>	(stellar_sdk.operation.ManageData	H
<code>class method</code>), 105		<code>Hash</code> (class in stellar_sdk.xdr.hash), 187
<code>from_xdr_object()</code>	(stellar_sdk.operation.ManageSellOffer	<code>hash()</code> (stellar_sdk.fee_bump_transaction_envelope.FeeBumpTransactionEnvelope class method), 134
<code>class method</code>), 106		<code>hash()</code> (stellar_sdk.transaction_envelope.TransactionEnvelope method), 132
<code>from_xdr_object()</code>	(stellar_sdk.operation.Operation	<code>hash_hex()</code> (stellar_sdk.fee_bump_transaction_envelope.FeeBumpTransactionEnvelope class method), 135
<code>class method</code>), 98		<code>hash_hex()</code> (stellar_sdk.transaction_envelope.TransactionEnvelope method), 132
<code>from_xdr_object()</code>	(stellar_sdk.operation.PathPaymentStrictReceive	<code>HashMemo</code> (class in stellar_sdk.memo), 95
<code>class method</code>), 106		<code>Hello</code> (class in stellar_sdk.xdr.hello), 187
<code>from_xdr_object()</code>	(stellar_sdk.operation.PathPaymentStrictSend	<code>HmacSha256Key</code> (class in stellar_sdk.xdr.hmac_sha256_key), 187
<code>class method</code>), 107		<code>HmacSha256Mac</code> (class in stellar_sdk.xdr.hmac_sha256_mac), 187
<code>from_xdr_object()</code>	(stellar_sdk.operation.Payment	<code>Hyper</code> (class in stellar_sdk.xdr.base), 188
<code>class method</code>), 107		I
<code>from_xdr_object()</code>	(stellar_sdk.operation.RevokeSponsorship	<code>IdMemo</code> (class in stellar_sdk.memo), 95
<code>class method</code>), 113		<code>include_failed()</code> (stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder method), 68
<code>from_xdr_object()</code>	(stellar_sdk.operation.SetOptions	
<code>class method</code>), 109		
<code>from_xdr_object()</code>	(stellar_sdk.operation.SetTrustLineFlags	
<code>class method</code>), 115		
<code>from_xdr_object()</code>	(stellar_sdk.price.Price	
<code>class method</code>), 116		
<code>from_xdr_object()</code>	(stellar_sdk.signer.Signer	
<code>class method</code>), 125		
<code>from_xdr_object()</code>	(stellar_sdk.signer_key.SignerKey	
<code>class method</code>), 126		
<code>from_xdr_object()</code>	(stellar_sdk.time_bounds.TimeBounds	
<code>class</code>		

include_failed() (stellar_sdk.call_builder.call_builder_async.PaymentsCallBuilder method), 71

include_failed() (stellar_sdk.call_builder.call_builder_async.TransactionsCallBuilder method), 81

include_failed() (stellar_sdk.call_builder.call_builder_sync.OperationsCallBuilder method), 39

include_failed() (stellar_sdk.call_builder.call_builder_sync.PaymentsCallBuilder method), 42

include_failed() (stellar_sdk.call_builder.call_builder_sync.TransactionsCallBuilder method), 52

increment_sequence_number() (stellar_sdk.account.Account method), 21

Inflation (class in stellar_sdk.operation), 102

InflationPayout (class in stellar_sdk.xdr.inflation_payout), 188

InflationResult (class in stellar_sdk.xdr.inflation_result), 188

InflationResultCode (class in stellar_sdk.xdr.inflation_result_code), 188

InnerTransactionResult (class in stellar_sdk.xdr.inner_transaction_result), 189

InnerTransactionResultExt (class in stellar_sdk.xdr.inner_transaction_result_ext), 190

InnerTransactionResultPair (class in stellar_sdk.xdr.inner_transaction_result_pair), 190

InnerTransactionResultResult (class in stellar_sdk.xdr.inner_transaction_result_result), 190

Int32 (class in stellar_sdk.xdr.int32), 191

Int64 (class in stellar_sdk.xdr.int64), 191

Integer (class in stellar_sdk.xdr.base), 191

InvalidFederationAddress (class in stellar_sdk.sep.exceptions), 159

InvalidSep10ChallengeError (class in stellar_sdk.sep.exceptions), 159

IPAddrType (class in stellar_sdk.xdr.ip_addr_type), 188

is_native() (stellar_sdk.asset.Asset method), 23

is_valid_ed25519_public_key() (stellar_sdk.strkey.StrKey static method), 128

is_valid_ed25519_secret_seed() (stellar_sdk.strkey.StrKey static method), 128

is_valid_lexicographic_order() (stellar_sdk.liquidity_pool_asset.LiquidityPoolAsset static method), 92

is_valid_pre_auth_tx() (stellar_sdk.strkey.StrKey static method), 128

is_valid_sha256_hash() (stellar_sdk.strkey.StrKey static method), 129

ITALIAN (stellar_sdk.sep.mnemonic.Language attribute), 152

JAPANESE (stellar_sdk.sep.mnemonic.Language attribute), 152

join() (stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder method), 68

join() (stellar_sdk.call_builder.call_builder_async.PaymentsCallBuilder method), 71

join() (stellar_sdk.call_builder.call_builder_sync.OperationsCallBuilder method), 39

join() (stellar_sdk.call_builder.call_builder_sync.PaymentsCallBuilder method), 42

json() (stellar_sdk.client.response.Response method), 87

K

Keypair (class in stellar_sdk.keypair), 89

KOREAN (stellar_sdk.sep.mnemonic.Language attribute), 152

L

Language (class in stellar_sdk.sep.mnemonic), 152

ledger() (stellar_sdk.call_builder.call_builder_async.LedgersCallBuilder method), 62

ledger() (stellar_sdk.call_builder.call_builder_sync.LedgersCallBuilder method), 33

LedgerCloseMeta (class in stellar_sdk.xdr.ledger_close_meta), 191

LedgerCloseMetaV0 (class in stellar_sdk.xdr.ledger_close_meta_v0), 191

LedgerCloseValueSignature (class in stellar_sdk.xdr.ledger_close_value_signature), 192

LedgerEntry (class in stellar_sdk.xdr.ledger_entry), 192

LedgerEntryChange (class in stellar_sdk.xdr.ledger_entry_change), 193

LedgerEntryChanges (class in stellar_sdk.xdr.ledger_entry_changes), 193

LedgerEntryChangeType (class in stellar_sdk.xdr.ledger_entry_change_type), 193

LedgerEntryData (class in stellar_sdk.xdr.ledger_entry_data), 194

LedgerEntryExt (class in stellar_sdk.xdr.ledger_entry_ext), 194

LedgerEntryExtensionV1 (class in stellar_sdk.xdr.ledger_entry_extension_v1), 194

LedgerEntryExtensionV1Ext (class in stellar_sdk.xdr.ledger_entry_extension_v1_ext), 195

LedgerEntryType	(class in stellar_sdk.xdr.ledger_entry_type), 195	limit()	(stellar_sdk.call_builder.call_builder_async.LedgersCallBuilder method), 63
LedgerHeader	(class in stellar_sdk.xdr.ledger_header), 195	limit()	(stellar_sdk.call_builder.call_builder_async.LiquidityPoolsBuilder method), 64
LedgerHeaderExt	(class in stellar_sdk.xdr.ledger_header_ext), 196	limit()	(stellar_sdk.call_builder.call_builder_async.OffersCallBuilder method), 66
LedgerHeaderHistoryEntry	(class in stellar_sdk.xdr.ledger_header_history_entry), 196	limit()	(stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder method), 68
LedgerHeaderHistoryEntryExt	(class in stellar_sdk.xdr.ledger_header_history_entry_ext), 197	limit()	(stellar_sdk.call_builder.call_builder_async.OrderbookCallBuilder method), 70
LedgerKey	(class in stellar_sdk.xdr.ledger_key), 197	limit()	(stellar_sdk.call_builder.call_builder_async.PaymentsCallBuilder method), 71
LedgerKeyAccount	(class in stellar_sdk.xdr.ledger_key_account), 198	limit()	(stellar_sdk.call_builder.call_builder_async.RootCallBuilder method), 72
LedgerKeyClaimableBalance	(class in stellar_sdk.xdr.ledger_key_claimable_balance), 198	limit()	(stellar_sdk.call_builder.call_builder_async.StrictReceivePathsCallBuilder method), 74
LedgerKeyData	(class in stellar_sdk.xdr.ledger_key_data), 198	limit()	(stellar_sdk.call_builder.call_builder_async.StrictSendPathsCallBuilder method), 76
LedgerKeyLiquidityPool	(class in stellar_sdk.xdr.ledger_key_liquidity_pool), 199	limit()	(stellar_sdk.call_builder.call_builder_async.TradeAggregationsCallBuilder method), 77
LedgerKeyOffer	(class in stellar_sdk.xdr.ledger_key_offer), 199	limit()	(stellar_sdk.call_builder.call_builder_async.TradesCallBuilder method), 79
LedgerKeyTrustLine	(class in stellar_sdk.xdr.ledger_key_trust_line), 199	limit()	(stellar_sdk.call_builder.call_builder_async.TransactionsCallBuilder method), 81
ledgers()	(stellar_sdk.server.Server method), 117	limit()	(stellar_sdk.call_builder.call_builder_sync.AccountsCallBuilder method), 25
ledgers()	(stellar_sdk.server_async.ServerAsync method), 121	limit()	(stellar_sdk.call_builder.call_builder_sync.AssetsCallBuilder method), 26
LedgersCallBuilder	(class in stellar_sdk.call_builder.call_builder_async), 62	limit()	(stellar_sdk.call_builder.call_builder_sync.ClaimableBalancesCallBuilder method), 28
LedgersCallBuilder	(class in stellar_sdk.call_builder.call_builder_sync), 33	limit()	(stellar_sdk.call_builder.call_builder_sync.DataCallBuilder method), 29
LedgerSCPMessages	(class in stellar_sdk.xdr.ledger_scp_messages), 199	limit()	(stellar_sdk.call_builder.call_builder_sync.EffectsCallBuilder method), 31
LedgerUpgrade	(class in stellar_sdk.xdr.ledger_upgrade), 200	limit()	(stellar_sdk.call_builder.call_builder_sync.FeeStatsCallBuilder method), 32
LedgerUpgradeType	(class in stellar_sdk.xdr.ledger_upgrade_type), 200	limit()	(stellar_sdk.call_builder.call_builder_sync.LedgersCallBuilder method), 34
Liabilities	(class in stellar_sdk.xdr.liabilities), 200	limit()	(stellar_sdk.call_builder.call_builder_sync.LiquidityPoolsBuilder method), 35
limit()	(stellar_sdk.call_builder.call_builder_async.AccountsCallBuilder method), 54	limit()	(stellar_sdk.call_builder.call_builder_sync.OffersCallBuilder method), 37
limit()	(stellar_sdk.call_builder.call_builder_async.AssetsCallBuilder method), 55	limit()	(stellar_sdk.call_builder.call_builder_sync.OperationsCallBuilder method), 39
limit()	(stellar_sdk.call_builder.call_builder_async.ClaimableBalancesCallBuilder method), 57	limit()	(stellar_sdk.call_builder.call_builder_sync.OrderbookCallBuilder method), 41
limit()	(stellar_sdk.call_builder.call_builder_async.DataCallBuilder method), 58	limit()	(stellar_sdk.call_builder.call_builder_sync.PaymentsCallBuilder method), 42
limit()	(stellar_sdk.call_builder.call_builder_async.EffectsCallBuilder method), 60	limit()	(stellar_sdk.call_builder.call_builder_sync.RootCallBuilder method), 43
limit()	(stellar_sdk.call_builder.call_builder_async.FeeStatsCallBuilder method), 61	limit()	(stellar_sdk.call_builder.call_builder_sync.StrictReceivePathsCallBuilder method), 45
		limit()	(stellar_sdk.call_builder.call_builder_sync.StrictSendPathsCallBuilder method), 47

[limit\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.TradeAggregationCallBuilder* method), 48
[limit\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.TradesCallBuilder* method), 50
[limit\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.TransactionBuilder* method), 52
[liquidity_pool\(\)](#) (*stellar_sdk.call_builder.call_builder_async.LiquidityPoolsBuilder* method), 64
[liquidity_pool\(\)](#) (*stellar_sdk.call_builder.call_builder_sync.LiquidityPoolsBuilder* method), 35
[LIQUIDITY_POOL_FEE_V18](#) (in module *stellar_sdk.liquidity_pool_asset*), 92
[LIQUIDITY_POOL_FEE_V18](#) (in module *stellar_sdk.xdr.constants*), 256
[liquidity_pool_id](#) (*stellar_sdk.liquidity_pool_asset.LiquidityPoolAsset* property), 93
[liquidity_pools\(\)](#) (*stellar_sdk.server.Server* method), 117
[liquidity_pools\(\)](#) (*stellar_sdk.server_async.ServerAsync* method), 122
[LiquidityPoolAsset](#) (class in *stellar_sdk.liquidity_pool_asset*), 92
[LiquidityPoolConstantProductParameters](#) (class in *stellar_sdk.xdr.liquidity_pool_constant_product_parameters*), 201
[LiquidityPoolDeposit](#) (class in *stellar_sdk.operation*), 103
[LiquidityPoolDepositOp](#) (class in *stellar_sdk.xdr.liquidity_pool_deposit_op*), 201
[LiquidityPoolDepositResult](#) (class in *stellar_sdk.xdr.liquidity_pool_deposit_result*), 201
[LiquidityPoolDepositResultCode](#) (class in *stellar_sdk.xdr.liquidity_pool_deposit_result_code*), 202
[LiquidityPoolEntry](#) (class in *stellar_sdk.xdr.liquidity_pool_entry*), 202
[LiquidityPoolEntryBody](#) (class in *stellar_sdk.xdr.liquidity_pool_entry_body*), 203
[LiquidityPoolEntryConstantProduct](#) (class in *stellar_sdk.xdr.liquidity_pool_entry_constant_product*), 203
[LiquidityPoolId](#) (class in *stellar_sdk.liquidity_pool_id*), 93
[LiquidityPoolParameters](#) (class in *stellar_sdk.xdr.liquidity_pool_parameters*), 204
[LiquidityPoolsBuilder](#) (class in *stellar_sdk.call_builder.call_builder_async*), 63
[LiquidityPoolsBuilder](#) (class in *stellar_sdk.call_builder.call_builder_sync*), 34
[LiquidityPoolType](#) (class in *stellar_sdk.xdr.liquidity_pool_type*), 204
[LiquidityPoolWithdraw](#) (class in *stellar_sdk.operation*), 103
[LiquidityPoolWithdrawOp](#) (class in *stellar_sdk.xdr.liquidity_pool_withdraw_op*), 204
[LiquidityPoolWithdrawResult](#) (class in *stellar_sdk.xdr.liquidity_pool_withdraw_result*), 204
[LiquidityPoolWithdrawResultCode](#) (class in *stellar_sdk.xdr.liquidity_pool_withdraw_result_code*), 205
[load_account\(\)](#) (*stellar_sdk.server.Server* method), 118
[load_account\(\)](#) (*stellar_sdk.server_async.ServerAsync* method), 122
[load_ed25519_public_key_signers\(\)](#) (*stellar_sdk.account.Account* method), 21

M

[ManageBuyOffer](#) (class in *stellar_sdk.operation*), 104
[ManageBuyOfferOp](#) (class in *stellar_sdk.xdr.manage_buy_offer_op*), 205
[ManageBuyOfferResult](#) (class in *stellar_sdk.xdr.manage_buy_offer_result*), 205
[ManageBuyOfferResultCode](#) (class in *stellar_sdk.xdr.manage_buy_offer_result_code*), 206
[ManageData](#) (class in *stellar_sdk.operation*), 105
[ManageDataOp](#) (class in *stellar_sdk.xdr.manage_data_op*), 206
[ManageDataResult](#) (class in *stellar_sdk.xdr.manage_data_result*), 206
[ManageDataResultCode](#) (class in *stellar_sdk.xdr.manage_data_result_code*), 207
[ManageOfferEffect](#) (class in *stellar_sdk.xdr.manage_offer_effect*), 207
[ManageOfferSuccessResult](#) (class in *stellar_sdk.xdr.manage_offer_success_result*), 207
[ManageOfferSuccessResultOffer](#) (class in *stellar_sdk.xdr.manage_offer_success_result_offer*), 208
[ManageSellOffer](#) (class in *stellar_sdk.operation*), 105
[ManageSellOfferOp](#) (class in *stellar_sdk.xdr.manage_sell_offer_op*), 208
[ManageSellOfferResult](#) (class in *stellar_sdk.xdr.manage_sell_offer_result*), 208

ManageSellOfferResultCode (class in stellar_sdk.xdr.manage_sell_offer_result_code), 209
MASK_ACCOUNT_FLAGS (in module stellar_sdk.xdr.constants), 256
MASK_ACCOUNT_FLAGS_V17 (in module stellar_sdk.xdr.constants), 256
MASK_CLAIMABLE_BALANCE_FLAGS (in module stellar_sdk.xdr.constants), 256
MASK_OFFERENTRY_FLAGS (in module stellar_sdk.xdr.constants), 256
MASK_TRUSTLINE_FLAGS (in module stellar_sdk.xdr.constants), 256
MASK_TRUSTLINE_FLAGS_V13 (in module stellar_sdk.xdr.constants), 256
MASK_TRUSTLINE_FLAGS_V17 (in module stellar_sdk.xdr.constants), 256
MAX_OPS_PER_TX (in module stellar_sdk.xdr.constants), 256
MAX_SIGNERS (in module stellar_sdk.xdr.constants), 256
Memo (class in stellar_sdk.memo), 94
Memo (class in stellar_sdk.xdr.memo), 209
MemoInvalidException (class in stellar_sdk.exceptions), 88
MemoType (class in stellar_sdk.xdr.memo_type), 210
MessageType (class in stellar_sdk.xdr.message_type), 210
MissingEd25519SecretSeedError (class in stellar_sdk.exceptions), 88
module
 stellar_sdk, 21
MuxedAccount (class in stellar_sdk.muxed_account), 96
MuxedAccount (class in stellar_sdk.xdr.muxed_account), 211
MuxedAccountMed25519 (class in stellar_sdk.xdr.muxed_account_med25519), 211
N
native() (stellar_sdk.asset.Asset class method), 23
Network (class in stellar_sdk.network), 97
network_id() (stellar_sdk.network.Network method), 97
NoApproximationError (class in stellar_sdk.exceptions), 88
NodeID (class in stellar_sdk.xdr.node_id), 211
NoneMemo (class in stellar_sdk.memo), 94
NotFoundError (class in stellar_sdk.exceptions), 89
O
Offer (class in stellar_sdk.operation.revoke_sponsorship), 113
offer() (stellar_sdk.call_builder.call_builder_async.OfferCallBuilder method), 66
offer() (stellar_sdk.call_builder.call_builder_sync.OfferCallBuilder method), 37
OfferEntry (class in stellar_sdk.xdr.offer_entry), 211
OfferEntryExt (class in stellar_sdk.xdr.offer_entry_ext), 212
OfferEntryFlags (class in stellar_sdk.xdr.offer_entry_flags), 212
offers() (stellar_sdk.server.Server method), 118
offers() (stellar_sdk.server_async.ServerAsync method), 122
OffersCallBuilder (class in stellar_sdk.call_builder.call_builder_async), 65
OffersCallBuilder (class in stellar_sdk.call_builder.call_builder_sync), 36
Opaque (class in stellar_sdk.xdr.base), 212
Operation (class in stellar_sdk.operation), 98
Operation (class in stellar_sdk.xdr.operation), 213
operation() (stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder method), 68
operation() (stellar_sdk.call_builder.call_builder_sync.OperationsCallBuilder method), 39
OperationBody (class in stellar_sdk.xdr.operation_body), 214
OperationID (class in stellar_sdk.xdr.operation_id), 215
OperationIDId (class in stellar_sdk.xdr.operation_id_id), 216
OperationMeta (class in stellar_sdk.xdr.operation_meta), 216
OperationResult (class in stellar_sdk.xdr.operation_result), 216
OperationResultCode (class in stellar_sdk.xdr.operation_result_code), 217
OperationResultTr (class in stellar_sdk.xdr.operation_result_tr), 218
operations() (stellar_sdk.server.Server method), 118
operations() (stellar_sdk.server_async.ServerAsync method), 122
OperationsCallBuilder (class in stellar_sdk.call_builder.call_builder_async), 67
OperationsCallBuilder (class in stellar_sdk.call_builder.call_builder_sync), 38
OperationType (class in stellar_sdk.xdr.operation_type), 219
order() (stellar_sdk.call_builder.call_builder_async.AccountsCallBuilder method), 54
order() (stellar_sdk.call_builder.call_builder_async.AssetsCallBuilder method), 56
order() (stellar_sdk.call_builder.call_builder_async.ClaimableBalancesCallBuilder method), 57

- order() (stellar_sdk.call_builder.call_builder_async.DataCallBuilder method), 58
- order() (stellar_sdk.call_builder.call_builder_async.EffectsCallBuilder method), 60
- order() (stellar_sdk.call_builder.call_builder_async.FeeStatsCallBuilder method), 61
- order() (stellar_sdk.call_builder.call_builder_async.LedgersCallBuilder method), 63
- order() (stellar_sdk.call_builder.call_builder_async.LiquidityPoolsCallBuilder method), 64
- order() (stellar_sdk.call_builder.call_builder_async.OfferCallBuilder method), 66
- order() (stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder method), 68
- order() (stellar_sdk.call_builder.call_builder_async.OrderbookCallBuilder method), 70
- order() (stellar_sdk.call_builder.call_builder_async.PaymentsCallBuilder method), 71
- order() (stellar_sdk.call_builder.call_builder_async.RootOrderbookCallBuilder method), 73
- order() (stellar_sdk.call_builder.call_builder_async.StrictReceivePathsCallBuilder method), 74
- order() (stellar_sdk.call_builder.call_builder_async.StrictSendPathsCallBuilder method), 76
- order() (stellar_sdk.call_builder.call_builder_async.TradeAggregationsCallBuilder method), 77
- order() (stellar_sdk.call_builder.call_builder_async.TradesCallBuilder method), 79
- order() (stellar_sdk.call_builder.call_builder_async.TransactionsCallBuilder method), 81
- order() (stellar_sdk.call_builder.call_builder_sync.AccountCallBuilder method), 25
- order() (stellar_sdk.call_builder.call_builder_sync.AssetsCallBuilder method), 27
- order() (stellar_sdk.call_builder.call_builder_sync.ClaimableBalanceCallBuilder method), 28
- order() (stellar_sdk.call_builder.call_builder_sync.DataCallBuilder method), 29
- order() (stellar_sdk.call_builder.call_builder_sync.EffectsCallBuilder method), 31
- order() (stellar_sdk.call_builder.call_builder_sync.FeeStatsCallBuilder method), 32
- order() (stellar_sdk.call_builder.call_builder_sync.LedgersCallBuilder method), 34
- order() (stellar_sdk.call_builder.call_builder_sync.LiquidityPoolsCallBuilder method), 35
- order() (stellar_sdk.call_builder.call_builder_sync.OfferCallBuilder method), 37
- order() (stellar_sdk.call_builder.call_builder_sync.OperationsCallBuilder method), 39
- order() (stellar_sdk.call_builder.call_builder_sync.OrderbookCallBuilder method), 41
- order() (stellar_sdk.call_builder.call_builder_sync.PaymentsCallBuilder method), 42
- order() (stellar_sdk.call_builder.call_builder_sync.RootCallBuilder method), 44
- order() (stellar_sdk.call_builder.call_builder_sync.StrictReceivePathsCallBuilder method), 45
- order() (stellar_sdk.call_builder.call_builder_sync.StrictSendPathsCallBuilder method), 47
- order() (stellar_sdk.call_builder.call_builder_sync.TradeAggregationsCallBuilder method), 48
- order() (stellar_sdk.call_builder.call_builder_sync.TradesCallBuilder method), 50
- order() (stellar_sdk.call_builder.call_builder_sync.TransactionsCallBuilder method), 52
- orderbook() (stellar_sdk.server.Server method), 118
- orderbook() (stellar_sdk.server_async.ServerAsync method), 122
- OrderbookCallBuilder (class in stellar_sdk.call_builder.call_builder_async), 69
- OrderbookCallBuilder (class in stellar_sdk.call_builder.call_builder_sync), 69
- PathPaymentStrictReceive (class in stellar_sdk.operation), 106
- PathPaymentStrictReceiveOp (class in stellar_sdk.xdr.path_payment_strict_receive_op), 220
- PathPaymentStrictReceiveResult (class in stellar_sdk.xdr.path_payment_strict_receive_result), 221
- PathPaymentStrictReceiveResultCode (class in stellar_sdk.xdr.path_payment_strict_receive_result_code), 221
- PathPaymentStrictReceiveResultSuccess (class in stellar_sdk.xdr.path_payment_strict_receive_result_success), 222
- PathPaymentStrictSend (class in stellar_sdk.operation), 107
- PathPaymentStrictSendOp (class in stellar_sdk.xdr.path_payment_strict_send_op), 222
- PathPaymentStrictSendResult (class in stellar_sdk.xdr.path_payment_strict_send_result), 223
- PathPaymentStrictSendResultCode (class in stellar_sdk.xdr.path_payment_strict_send_result_code), 223
- PathPaymentStrictSendResultSuccess (class in stellar_sdk.xdr.path_payment_strict_send_result_success), 223

lar_sdk.xdr.path_payment_strict_send_result_success), class method), 111
 224
 Payment (class in *stellar_sdk.operation*), 107
 PaymentOp (class in *stellar_sdk.xdr.payment_op*), 224
 PaymentResult (class in *stellar_sdk.xdr.payment_result*), 224
 PaymentResultCode (class in *stellar_sdk.xdr.payment_result_code*), 225
 payments() (*stellar_sdk.server.Server* method), 118
 payments() (*stellar_sdk.server_async.ServerAsync* method), 122
 PaymentsCallBuilder (class in *stellar_sdk.call_builder.call_builder_async*), 70
 PaymentsCallBuilder (class in *stellar_sdk.call_builder.call_builder_sync*), 41
 PayStellarUri (class in *stellar_sdk.sep.stellar_uri*), 152
 PeerAddress (class in *stellar_sdk.xdr.peer_address*), 225
 PeerAddressIp (class in *stellar_sdk.xdr.peer_address_ip*), 225
 PeerStatList (class in *stellar_sdk.xdr.peer_stat_list*), 226
 PeerStats (class in *stellar_sdk.xdr.peer_stats*), 226
 PoolID (class in *stellar_sdk.xdr.pool_id*), 227
 post() (*stellar_sdk.client.aiohttp_client.AiohttpClient* method), 84
 post() (*stellar_sdk.client.base_async_client.BaseAsyncClient* method), 82
 post() (*stellar_sdk.client.base_sync_client.BaseSyncClient* method), 83
 post() (*stellar_sdk.client.requests_client.RequestsClient* method), 85
 post() (*stellar_sdk.client.simple_requests_client.SimpleRequestsClient* method), 86
 pre_auth_tx() (*stellar_sdk.signer.Signer* class method), 125
 pre_auth_tx() (*stellar_sdk.signer_key.SignerKey* class method), 126
 predicate_and() (*stellar_sdk.operation.ClaimPredicate* class method), 110
 predicate_before_absolute_time() (*stellar_sdk.operation.ClaimPredicate* class method), 110
 predicate_before_relative_time() (*stellar_sdk.operation.ClaimPredicate* class method), 110
 predicate_not() (*stellar_sdk.operation.ClaimPredicate* class method), 110
 predicate_or() (*stellar_sdk.operation.ClaimPredicate* class method), 111
 predicate_unconditional() (*stellar_sdk.operation.ClaimPredicate* class method), 111
 Price (class in *stellar_sdk.price*), 115
 Price (class in *stellar_sdk.xdr.price*), 227
 public_key (*stellar_sdk.keypair.Keypair* property), 91
 public_network() (*stellar_sdk.network.Network* class method), 97
 PUBLIC_NETWORK_PASSPHRASE (*stellar_sdk.network.Network* attribute), 97
 PublicKey (class in *stellar_sdk.xdr.public_key*), 227
 PublicKeyType (class in *stellar_sdk.xdr.public_key_type*), 227
R
 random() (*stellar_sdk.keypair.Keypair* class method), 91
 raw_public_key() (*stellar_sdk.keypair.Keypair* method), 91
 raw_secret_key() (*stellar_sdk.keypair.Keypair* method), 91
 read_challenge_transaction() (in module *stellar_sdk.sep.stellar_web_authentication*), 155
 Replacement (class in *stellar_sdk.sep.stellar_uri*), 154
 RequestsClient (class in *stellar_sdk.client.requests_client*), 85
 resolve_account_id() (in module *stellar_sdk.sep.federation*), 151
 resolve_account_id_async() (in module *stellar_sdk.sep.federation*), 151
 resolve_stellar_address() (in module *stellar_sdk.sep.federation*), 150
 resolve_stellar_address_async() (in module *stellar_sdk.sep.federation*), 151
 Response (class in *stellar_sdk.client.response*), 87
 ReturnHashMemo (class in *stellar_sdk.memo*), 96
 RevokeSponsorship (class in *stellar_sdk.operation*), 113
 RevokeSponsorshipOp (class in *stellar_sdk.xdr.revoke_sponsorship_op*), 227
 RevokeSponsorshipOpSigner (class in *stellar_sdk.xdr.revoke_sponsorship_op_signer*), 228
 RevokeSponsorshipResult (class in *stellar_sdk.xdr.revoke_sponsorship_result*), 228
 RevokeSponsorshipResultCode (class in *stellar_sdk.xdr.revoke_sponsorship_result_code*), 228
 RevokeSponsorshipType (class in *stellar_sdk.operation.revoke_sponsorship*), 113
 RevokeSponsorshipType (class in *stellar_sdk.xdr.revoke_sponsorship_type*), 229
 root() (*stellar_sdk.server.Server* method), 118

- root() (*stellar_sdk.server_async.ServerAsync* method), 123
- RootCallBuilder (class in *stellar_sdk.call_builder.call_builder_async*), 72
- RootCallBuilder (class in *stellar_sdk.call_builder.call_builder_sync*), 43
- ## S
- SCPBallot (class in *stellar_sdk.xdr.scp_ballot*), 229
- SCPEnvelope (class in *stellar_sdk.xdr.scp_envelope*), 229
- SCPHistoryEntry (class in *stellar_sdk.xdr.scp_history_entry*), 229
- SCPHistoryEntryV0 (class in *stellar_sdk.xdr.scp_history_entry_v0*), 230
- SCPNomination (class in *stellar_sdk.xdr.scp_nomination*), 230
- SCPQuorumSet (class in *stellar_sdk.xdr.scp_quorum_set*), 230
- SCPStatement (class in *stellar_sdk.xdr.scp_statement*), 230
- SCPStatementConfirm (class in *stellar_sdk.xdr.scp_statement_confirm*), 231
- SCPStatementExternalize (class in *stellar_sdk.xdr.scp_statement_externalize*), 232
- SCPStatementPledges (class in *stellar_sdk.xdr.scp_statement_pledges*), 232
- SCPStatementPrepare (class in *stellar_sdk.xdr.scp_statement_prepare*), 233
- SCPStatementType (class in *stellar_sdk.xdr.scp_statement_type*), 233
- SdkError (class in *stellar_sdk.exceptions*), 87
- secret (*stellar_sdk.keypair.Keypair* property), 91
- SequenceNumber (class in *stellar_sdk.xdr.sequence_number*), 233
- Server (class in *stellar_sdk.server*), 116
- ServerAsync (class in *stellar_sdk.server_async*), 120
- set_timeout() (*stellar_sdk.transaction_builder.TransactionBuilder* method), 148
- SetOptions (class in *stellar_sdk.operation*), 108
- SetOptionsOp (class in *stellar_sdk.xdr.set_options_op*), 233
- SetOptionsResult (class in *stellar_sdk.xdr.set_options_result*), 234
- SetOptionsResultCode (class in *stellar_sdk.xdr.set_options_result_code*), 234
- SetTrustLineFlags (class in *stellar_sdk.operation*), 115
- SetTrustLineFlagsOp (class in *stellar_sdk.xdr.set_trust_line_flags_op*), 235
- SetTrustLineFlagsResult (class in *stellar_sdk.xdr.set_trust_line_flags_result*), 235
- SetTrustLineFlagsResultCode (class in *stellar_sdk.xdr.set_trust_line_flags_result_code*), 235
- sha256_hash() (*stellar_sdk.signer.Signer* class method), 125
- sha256_hash() (*stellar_sdk.signer_key.SignerKey* class method), 126
- sign() (*stellar_sdk.fee_bump_transaction_envelope.FeeBumpTransactionEnvelope* method), 135
- sign() (*stellar_sdk.keypair.Keypair* method), 91
- sign() (*stellar_sdk.sep.stellar_uri.PayStellarUri* method), 153
- sign() (*stellar_sdk.sep.stellar_uri.TransactionStellarUri* method), 154
- sign() (*stellar_sdk.transaction_envelope.TransactionEnvelope* method), 132
- sign_decorated() (*stellar_sdk.keypair.Keypair* method), 91
- sign_hashx() (*stellar_sdk.fee_bump_transaction_envelope.FeeBumpTransactionEnvelope* method), 135
- sign_hashx() (*stellar_sdk.transaction_envelope.TransactionEnvelope* method), 132
- Signature (class in *stellar_sdk.xdr.signature*), 236
- signature_base() (*stellar_sdk.fee_bump_transaction_envelope.FeeBumpTransactionEnvelope* method), 135
- signature_base() (*stellar_sdk.transaction_envelope.TransactionEnvelope* method), 132
- signature_hint() (*stellar_sdk.keypair.Keypair* method), 92
- SignatureExistError (class in *stellar_sdk.exceptions*), 88
- SignatureHint (class in *stellar_sdk.xdr.signature_hint*), 236
- SignedSurveyRequestMessage (class in *stellar_sdk.xdr.signed_survey_request_message*), 236
- SignedSurveyResponseMessage (class in *stellar_sdk.xdr.signed_survey_response_message*), 236
- Signer (class in *stellar_sdk.operation.revoke_sponsorship*), 113
- Signer (class in *stellar_sdk.signer*), 124
- Signer (class in *stellar_sdk.xdr.signer*), 236
- SignerKey (class in *stellar_sdk.signer_key*), 126
- SignerKey (class in *stellar_sdk.xdr.signer_key*), 237
- SignerKeyType (class in *stellar_sdk.signer_key*), 126
- SignerKeyType (class in *stellar_sdk.xdr.signer_key_type*), 237
- SimplePaymentResult (class in *stellar_sdk.xdr.simple_payment_result*), 237
- SimpleRequestsClient (class in *stellar_sdk.client.simple_requests_client*), 86

SPANISH (<i>stellar_sdk.sep.mnemonic.Language</i> attribute), 152	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.AccountsCallBuilder</i> method), 25
<code>SponsorshipDescriptor</code> (class in <i>stellar_sdk.xdr.sponsorship_descriptor</i>), 237	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.AssetsCallBuilder</i> method), 27
<code>stellar_sdk</code> module, 21	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.ClaimableBalancesCallBuilder</i> method), 28
<code>StellarMessage</code> (class in <i>stellar_sdk.xdr.stellar_message</i>), 238	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.DataCallBuilder</i> method), 30
<code>StellarMnemonic</code> (class in <i>stellar_sdk.sep.mnemonic</i>), 152	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.EffectsCallBuilder</i> method), 31
<code>StellarTomlNotFoundError</code> (class in <i>stellar_sdk.sep.exceptions</i>), 159	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.FeeStatsCallBuilder</i> method), 33
<code>StellarValue</code> (class in <i>stellar_sdk.xdr.stellar_value</i>), 239	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.LedgersCallBuilder</i> method), 34
<code>StellarValueExt</code> (class in <i>stellar_sdk.xdr.stellar_value_ext</i>), 239	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.LiquidityPoolsBuilder</i> method), 35
<code>StellarValueType</code> (class in <i>stellar_sdk.xdr.stellar_value_type</i>), 239	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.OffersCallBuilder</i> method), 37
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.AccountsCallBuilder</i> method), 54	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.OperationsCallBuilder</i> method), 40
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.AssetsCallBuilder</i> method), 56	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.OrderbookCallBuilder</i> method), 41
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.ClaimableBalancesCallBuilder</i> method), 57	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.PaymentsCallBuilder</i> method), 43
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.DataCallBuilder</i> method), 59	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.RootCallBuilder</i> method), 44
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.EffectsCallBuilder</i> method), 60	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.StrictReceivePathsCallBuilder</i> method), 45
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.FeeStatsCallBuilder</i> method), 62	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.StrictSendPathsCallBuilder</i> method), 47
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.LedgersCallBuilder</i> method), 63	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.TradeAggregationsCallBuilder</i> method), 48
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.LiquidityPoolsBuilder</i> method), 64	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.TradesCallBuilder</i> method), 50
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.OffersCallBuilder</i> method), 66	<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_sync.TransactionsCallBuilder</i> method), 52
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.OperationsCallBuilder</i> method), 69	<code>stream()</code> (<i>stellar_sdk.client.aihttp_client.AiohttpClient</i> method), 84
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.OrderbookCallBuilder</i> method), 70	<code>stream()</code> (<i>stellar_sdk.client.base_async_client.BaseAsyncClient</i> method), 82
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.PaymentsCallBuilder</i> method), 72	<code>stream()</code> (<i>stellar_sdk.client.base_sync_client.BaseSyncClient</i> method), 83
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.RootCallBuilder</i> method), 73	<code>stream()</code> (<i>stellar_sdk.client.requests_client.RequestsClient</i> method), 85
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.StrictReceivePathsCallBuilder</i> method), 74	<code>stream()</code> (<i>stellar_sdk.client.simple_requests_client.SimpleRequestsClient</i> method), 86
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.StrictSendPathsCallBuilder</i> method), 76	<code>strict_receive_paths()</code> (<i>stellar_sdk.server.Server</i> method), 118
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.TradeAggregationsCallBuilder</i> method), 77	<code>strict_send_paths()</code> (<i>stellar_sdk.server_async.ServerAsync</i> method), 119
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.TradesCallBuilder</i> method), 79	<code>strict_send_paths()</code> (<i>stellar_sdk.server.Server</i> method), 119
<code>stream()</code> (<i>stellar_sdk.call_builder.call_builder_async.TransactionsCallBuilder</i> method), 81	

- lar_sdk.server_async.ServerAsync* method), 123
- StrictReceivePathsCallBuilder* (class in *stellar_sdk.call_builder.call_builder_async*), 73
- StrictReceivePathsCallBuilder* (class in *stellar_sdk.call_builder.call_builder_sync*), 44
- StrictSendPathsCallBuilder* (class in *stellar_sdk.call_builder.call_builder_async*), 75
- StrictSendPathsCallBuilder* (class in *stellar_sdk.call_builder.call_builder_sync*), 46
- String* (class in *stellar_sdk.xdr.base*), 240
- String32* (class in *stellar_sdk.xdr.string32*), 240
- String64* (class in *stellar_sdk.xdr.string64*), 240
- StrKey* (class in *stellar_sdk.strkey*), 127
- submit_transaction()* (*stellar_sdk.server.Server* method), 119
- submit_transaction()* (*stellar_sdk.server_async.ServerAsync* method), 123
- SurveyMessageCommandType* (class in *stellar_sdk.xdr.survey_message_command_type*), 240
- SurveyRequestMessage* (class in *stellar_sdk.xdr.survey_request_message*), 240
- SurveyResponseBody* (class in *stellar_sdk.xdr.survey_response_body*), 241
- SurveyResponseMessage* (class in *stellar_sdk.xdr.survey_response_message*), 241
- ## T
- testnet_network()* (*stellar_sdk.network.Network* class method), 98
- TESTNET_NETWORK_PASSPHRASE* (*stellar_sdk.network.Network* attribute), 97
- TextMemo* (class in *stellar_sdk.memo*), 95
- ThresholdIndexes* (class in *stellar_sdk.xdr.threshold_indexes*), 241
- Thresholds* (class in *stellar_sdk.xdr.thresholds*), 242
- TimeBounds* (class in *stellar_sdk.time_bounds*), 129
- TimeBounds* (class in *stellar_sdk.xdr.time_bounds*), 242
- TimePoint* (class in *stellar_sdk.xdr.time_point*), 242
- to_change_trust_asset_xdr_object()* (*stellar_sdk.asset.Asset* method), 23
- to_change_trust_asset_xdr_object()* (*stellar_sdk.liquidity_pool_asset.LiquidityPoolAsset* method), 93
- to_dict()* (*stellar_sdk.asset.Asset* method), 23
- to_transaction_envelope_v1()* (*stellar_sdk.transaction_envelope.TransactionEnvelope* method), 132
- to_trust_line_asset_xdr_object()* (*stellar_sdk.asset.Asset* method), 23
- to_trust_line_asset_xdr_object()* (*stellar_sdk.liquidity_pool_id.LiquidityPoolId* method), 93
- to_txrep()* (in module *stellar_sdk.sep.txrep*), 158
- to_uri()* (*stellar_sdk.sep.stellar_uri.PayStellarUri* method), 153
- to_uri()* (*stellar_sdk.sep.stellar_uri.TransactionStellarUri* method), 154
- to_xdr()* (*stellar_sdk.fee_bump_transaction_envelope.FeeBumpTransactionEnvelope* method), 135
- to_xdr()* (*stellar_sdk.transaction_envelope.TransactionEnvelope* method), 133
- to_xdr_amount()* (in module *stellar_sdk.xdr.utils*), 149
- to_xdr_amount()* (*stellar_sdk.operation.Operation* static method), 98
- to_xdr_object()* (*stellar_sdk.asset.Asset* method), 23
- to_xdr_object()* (*stellar_sdk.decorated_signature.DecoratedSignature* method), 129
- to_xdr_object()* (*stellar_sdk.fee_bump_transaction.FeeBumpTransaction* method), 134
- to_xdr_object()* (*stellar_sdk.fee_bump_transaction_envelope.FeeBumpTransactionEnvelope* method), 135
- to_xdr_object()* (*stellar_sdk.memo.HashMemo* method), 95
- to_xdr_object()* (*stellar_sdk.memo.IdMemo* method), 95
- to_xdr_object()* (*stellar_sdk.memo.Memo* method), 94
- to_xdr_object()* (*stellar_sdk.memo.NoneMemo* method), 94
- to_xdr_object()* (*stellar_sdk.memo.ReturnHashMemo* method), 96
- to_xdr_object()* (*stellar_sdk.memo.TextMemo* method), 95
- to_xdr_object()* (*stellar_sdk.muxed_account.MuxedAccount* method), 97
- to_xdr_object()* (*stellar_sdk.operation.AccountMerge* method), 99
- to_xdr_object()* (*stellar_sdk.operation.AllowTrust* method), 100
- to_xdr_object()* (*stellar_sdk.operation.BeginSponsoringFutureReserves* method), 112
- to_xdr_object()* (*stellar_sdk.operation.BumpSequence* method), 100
- to_xdr_object()* (*stellar_sdk.operation.ChangeTrust* method), 101

- to_xdr_object() (stellar_sdk.operation.ClaimClaimableBalance method), 111
- to_xdr_object() (stellar_sdk.operation.Clawback method), 114
- to_xdr_object() (stellar_sdk.operation.ClawbackClaimableBalance method), 114
- to_xdr_object() (stellar_sdk.operation.CreateAccount method), 101
- to_xdr_object() (stellar_sdk.operation.CreateClaimableBalance method), 109
- to_xdr_object() (stellar_sdk.operation.CreatePassiveSellOffer method), 102
- to_xdr_object() (stellar_sdk.operation.EndSponsoringFutureReserves method), 112
- to_xdr_object() (stellar_sdk.operation.Inflation method), 102
- to_xdr_object() (stellar_sdk.operation.LiquidityPoolDeposit method), 103
- to_xdr_object() (stellar_sdk.operation.LiquidityPoolWithdraw method), 104
- to_xdr_object() (stellar_sdk.operation.ManageBuyOffer method), 104
- to_xdr_object() (stellar_sdk.operation.ManageData method), 105
- to_xdr_object() (stellar_sdk.operation.ManageSellOffer method), 106
- to_xdr_object() (stellar_sdk.operation.Operation method), 99
- to_xdr_object() (stellar_sdk.operation.PathPaymentStrictReceive method), 106
- to_xdr_object() (stellar_sdk.operation.PathPaymentStrictSend method), 107
- to_xdr_object() (stellar_sdk.operation.Payment method), 108
- to_xdr_object() (stellar_sdk.operation.RevokeSponsorship method), 113
- to_xdr_object() (stellar_sdk.operation.SetOptions method), 109
- to_xdr_object() (stellar_sdk.operation.SetTrustLineFlags method), 115
- to_xdr_object() (stellar_sdk.price.Price method), 116
- to_xdr_object() (stellar_sdk.signer.Signer method), 125
- to_xdr_object() (stellar_sdk.signer_key.SignerKey method), 126
- to_xdr_object() (stellar_sdk.time_bounds.TimeBounds method), 129
- to_xdr_object() (stellar_sdk.transaction.Transaction method), 131
- to_xdr_object() (stellar_sdk.transaction_envelope.TransactionEnvelope method), 133
- TopologyResponseBody (class in stellar_sdk.xdr.topology_response_body), 242
- trade_aggregations() (stellar_sdk.server.Server method), 119
- trade_aggregations() (stellar_sdk.server_async.ServerAsync method), 123
- TradeAggregationsCallBuilder (class in stellar_sdk.call_builder.call_builder_async), 76
- TradeAggregationsCallBuilder (class in stellar_sdk.call_builder.call_builder_sync), 47
- trades() (stellar_sdk.server.Server method), 120
- trades() (stellar_sdk.server_async.ServerAsync method), 124
- TradesCallBuilder (class in stellar_sdk.call_builder.call_builder_async), 78
- TradesCallBuilder (class in stellar_sdk.call_builder.call_builder_sync), 49
- Transaction (class in stellar_sdk.transaction), 130
- Transaction (class in stellar_sdk.xdr.transaction), 243
- transaction() (stellar_sdk.call_builder.call_builder_async.Transactions method), 81
- transaction() (stellar_sdk.call_builder.call_builder_sync.Transactions method), 52
- TransactionBuilder (class in stellar_sdk.transaction_builder), 136
- TransactionEnvelope (class in stellar_sdk.transaction_envelope), 131
- TransactionEnvelope (class in stellar_sdk.xdr.transaction_envelope), 243
- TransactionExt (class in stellar_sdk.xdr.transaction_ext), 244
- TransactionHistoryEntry (class in stellar_sdk.xdr.transaction_history_entry), 244
- TransactionHistoryEntryExt (class in stellar_sdk.xdr.transaction_history_entry_ext), 244
- TransactionHistoryResultEntry (class in stel-

- lar_sdk.xdr.transaction_history_result_entry*), 245
- TransactionHistoryResultEntryExt** (class in *stellar_sdk.xdr.transaction_history_result_entry_ext*), 245
- TransactionMeta** (class in *stellar_sdk.xdr.transaction_meta*), 245
- TransactionMetaV1** (class in *stellar_sdk.xdr.transaction_meta_v1*), 246
- TransactionMetaV2** (class in *stellar_sdk.xdr.transaction_meta_v2*), 246
- TransactionResult** (class in *stellar_sdk.xdr.transaction_result*), 246
- TransactionResultCode** (class in *stellar_sdk.xdr.transaction_result_code*), 247
- TransactionResultExt** (class in *stellar_sdk.xdr.transaction_result_ext*), 247
- TransactionResultMeta** (class in *stellar_sdk.xdr.transaction_result_meta*), 248
- TransactionResultPair** (class in *stellar_sdk.xdr.transaction_result_pair*), 248
- TransactionResultResult** (class in *stellar_sdk.xdr.transaction_result_result*), 248
- TransactionResultSet** (class in *stellar_sdk.xdr.transaction_result_set*), 249
- transactions()** (*stellar_sdk.server.Server* method), 120
- transactions()** (*stellar_sdk.server_async.ServerAsync* method), 124
- TransactionsCallBuilder** (class in *stellar_sdk.call_builder.call_builder_async*), 80
- TransactionsCallBuilder** (class in *stellar_sdk.call_builder.call_builder_sync*), 51
- TransactionSet** (class in *stellar_sdk.xdr.transaction_set*), 249
- TransactionSignaturePayload** (class in *stellar_sdk.xdr.transaction_signature_payload*), 249
- TransactionSignaturePayloadTaggedTransaction** (class in *stellar_sdk.xdr.transaction_signature_payload_tagged_transaction*), 249
- TransactionStellarUri** (class in *stellar_sdk.sep.stellar_uri*), 153
- TransactionV0** (class in *stellar_sdk.xdr.transaction_v0*), 250
- TransactionV0Envelope** (class in *stellar_sdk.xdr.transaction_v0_envelope*), 250
- TransactionV0Ext** (class in *stellar_sdk.xdr.transaction_v0_ext*), 251
- TransactionV1Envelope** (class in *stellar_sdk.xdr.transaction_v1_envelope*), 251
- TrustLine** (class in *stellar_sdk.operation.revoke_sponsorship*), 113
- TrustLineAsset** (class in *stellar_sdk.xdr.trust_line_asset*), 251
- TrustLineEntry** (class in *stellar_sdk.xdr.trust_line_entry*), 252
- TrustLineEntryExt** (class in *stellar_sdk.xdr.trust_line_entry_ext*), 252
- TrustLineEntryExtensionV2** (class in *stellar_sdk.xdr.trust_line_entry_extension_v2*), 253
- TrustLineEntryExtensionV2Ext** (class in *stellar_sdk.xdr.trust_line_entry_extension_v2_ext*), 253
- TrustLineEntryFlag** (class in *stellar_sdk.operation.allow_trust*), 100
- TrustLineEntryV1** (class in *stellar_sdk.xdr.trust_line_entry_v1*), 254
- TrustLineEntryV1Ext** (class in *stellar_sdk.xdr.trust_line_entry_v1_ext*), 254
- TrustLineFlags** (class in *stellar_sdk.operation.set_trust_line_flags*), 115
- TrustLineFlags** (class in *stellar_sdk.xdr.trust_line_flags*), 254
- type** (*stellar_sdk.asset.Asset* property), 23
- ## U
- Uint256** (class in *stellar_sdk.xdr.uint256*), 255
- Uint32** (class in *stellar_sdk.xdr.uint32*), 255
- Uint64** (class in *stellar_sdk.xdr.uint64*), 255
- universal_account_id** (*stellar_sdk.account.Account* property), 22
- universal_account_id** (*stellar_sdk.muxed_account.MuxedAccount* property), 97
- UnsignedHyper** (class in *stellar_sdk.xdr.base*), 255
- UnsignedInteger** (class in *stellar_sdk.xdr.base*), 255
- UpgradeEntryMeta** (class in *stellar_sdk.xdr.upgrade_entry_meta*), 255
- UpgradeType** (class in *stellar_sdk.xdr.upgrade_type*), 256
- ## V
- Value** (class in *stellar_sdk.xdr.value*), 256
- verify()** (*stellar_sdk.keypair.Keypair* method), 92
- verify_challenge_transaction()** (in module *stellar_sdk.sep.stellar_web_authentication*), 157
- verify_challenge_transaction_signed_by_client_master_key()** (in module *stellar_sdk.sep.stellar_web_authentication*), 156
- verify_challenge_transaction_signers()** (in module *stellar_sdk.sep.stellar_web_authentication*), 156

lar_sdk.sep.stellar_web_authentication),
157
`verify_challenge_transaction_threshold()`
(in *module* *stel-*
lar_sdk.sep.stellar_web_authentication),
155

X

`xdr_public_key()` (*stellar_sdk.keypair.Keypair*
method), 92