Transfer balances programmatically on Avail DA

Transferring funds programmatically

You can transferAVAIL from one account to another programmatically by calling any one of these three extrinsics from thebalances pallet on an Avail node:

1. balances_transferKeepAlive
2. : Transfers funds from one account to another, but does not allow the sender balance
3. to dip below the existential deposit.
4. balances_transferAllowDeath
5. : Transfers funds from one account to another, and allows the sender balance to dip below
6. the existential deposit.
7. balances_transferAll
8. : Transfers all funds from one account to another.

EXISTENTIAL DEPOSIT

Only accounts with a balance equal to or greater than theexistential deposit are stored on the state trie. The current value of the existential deposit is0.000001 AVAIL . Any account whose balane dips below this amount is'reaped' .

Transferring funds usingbalances_transferKeepAlive

avail-js avail-rust avail-go AvailApps explorer 1. Insideyour-file-name.ts 2. , add the following code:

avail-js import { SDK , WaitFor , Keyring , BN } from

"avail-js-sdk"

const

main

=

async () => { const

providerEndpoint

=

"wss://turing-rpc.avail.so/ws" ; const

sdk

=

await

SDK .New (providerEndpoint)

const

Alice

=

'This is a random seed phrase please replace with your own' ; const

account

=

new

Keyring ({ type :

"sr25519" }) .addFromUri (Alice) const

dest

=

"5HGjWAeFDfFCWPsjFQdVV2Msvz2XtMktvgocEZcCj68kUMaw"

// Eve const

amount

=

new

BN (12) .mul (new

BN (10) .pow (new

BN ("18"))) // twelve Avail

const

result

=

await

sdk . tx . balances .transferKeepAlive (dest , amount ,

WaitFor .BlockInclusion , account) if (result .isErr) { console .log (result .reason) process .exit (1) }

console .log ("From="

+

result . event .from +

", To="

+

result . event .to +

", Amount="

+

result . event .amount) console .log ("TxHash="

+

result .txHash +

", BlockHash="

+

result .blockHash)

process .exit () } main () 1. Run the code using:

ts-node

your-file-name.ts Your response should look something like this:

Sample Response: From=5CqgQkrDcdg5QrtuxT3H7WszrqgrBMhdwRbmMVXQzc4VSiEg,
To=5DDY2yzh8uCysYFAiRSTeQVwtZSKNF49CkQkyPH852xvrYKk, Amount=10000000000000000000
TxHash=0x45a3ec18b96c2bff0d92d70ba5f0fa904b79a49610e845b72d16ccf7c094533d,
BlockHash=0x9fa20525f0db53e144ff595f00728611d60bd5d5e597f07b82123301067e90be

Transferring funds usingbalances_transferAllowDeath

avail-js avail-rust avail-go AvailApps explorer 1. Insideyour-file-name.ts 2. , add the following code:

avail-js import { SDK , WaitFor , Keyring , BN } from

"avail-js-sdk"

const

main

=

async () => { const

providerEndpoint

=

"wss://turing-rpc.avail.so/ws" ; const

sdk

=

await

SDK .New (providerEndpoint)

const

Alice

=

'This is a random seed phrase please replace with your own' ; const

account

=

new

Keyring ({ type :

"sr25519" }) .addFromUri (Alice) const

dest

=

"5HGjWAeFDfFCWPsjFQdVV2Msvz2XtMktvgocEZcCj68kUMaw"

// Eve const

amount

=

new

BN (12) .mul (new

BN (10) .pow (new

BN ("18"))) // twelve Avail

const

result

=

await

sdk . tx . balances .transferAllowDeath (dest , amount ,

WaitFor .BlockInclusion , account) if (result .isErr) { console .log (result .reason) process .exit (1) }

console .log ("From="

+

result . event .from +

", To="

+

result . event .to +

", Amount="

+

result . event .amount) console .log ("MaybeKilled="

+

result . event2 ?.account) console .log ("TxHash="

+

result .txHash +

", BlockHash="

+

result .blockHash)

process .exit () } main () 1. Run the code using:

ts-node

your-file-name.ts Your response should look something like this:

Sample Response: From=5CqgQkrDcdg5QrtuxT3H7WszrqgrBMhdwRbmMVXQzc4VSiEg, To=5HGjWAeFDfFCWPsjFQdVV2Msvz2XtMktvgocEZcCj68kUMaw, Amount=12000000000000000000 MaybeKilled=undefined TxHash=0x4be89fffca3b7849066c3a7b6b29af318caa49ca12a1ba17610b0a30d97fd30e, BlockHash=0x861531ee9512849cec0bde5294bb65098424c84e3ab64b6c25722574370d8224

Transferring funds usingbalances_transferAll

avail-js avail-rust avail-go AvailApps explorer 1. Insideyour-file-name.ts 2. , add the following code:

avail-js import { SDK , WaitFor , Keyring , BN } from

"avail-js-sdk"

const

main

=

async () => { const

providerEndpoint

=

"wss://turing-rpc.avail.so/ws" ; const

sdk

=

await

SDK .New (providerEndpoint)

const

Alice

=

'This is a random seed phrase please replace with your own' ; const

account

=

new

Keyring ({ type :

"sr25519" }) .addFromUri (Alice) const

dest

=

"5HGjWAeFDfFCWPsjFQdVV2Msvz2XtMktvgocEZcCj68kUMaw"

// Eve const

keepAlive

=

false

const

result

=

await

sdk . tx . balances .transferAll (dest , keepAlive ,

WaitFor .BlockInclusion , account) if (result .isErr) { console .log (result .reason) process .exit (1) }

console .log ("From="

+

result . event .from +

", To="

+

result . event .to +

", Amount="

+

result . event .amount) console .log ("MaybeKilled="

+

result . event2 ?.account) console .log ("TxHash="

+

result .txHash +

", BlockHash="

+

result .blockHash)

process .exit () } main () 1. Run the code using:

ts-node

your-file-name.ts Your response should look something like this:

Sample Response: From=5CqgQkrDcdg5QrtuxT3H7WszrqgrBMhdwRbmMVXQzc4VSiEg, To=5DDY2yzh8uCysYFAiRSTeQVwtZSKNF49CkQkyPH852xvrYKk, Amount=14967497534555492726990 TxHash=0x3dbf0865aae15137e1fe3f922d70b7ff514a8c27e712d12ea1a8a1d4a7af9437, BlockHash=0x129745d18065f66f106b3744fe73ab5f9a1d7cb6205b271d13119399d3a56d31 If you check the balance of the sender account now, it will either be0 or equal to theexistential deposit amount, if the account is unable to be reaped due to dependencies on the network. In this case, while the account will continue to exist on the state trie, it's balance will be too low to perform any operations.

Query Balances on Avail DA Deploy a Rollup on Avail DA