

Bank Precompile

[illegible]

This precompile enables EVM clients to interact with the bank module.

Functions

Transactions

- send
- : Enables a user to send non-native tokens to another wallet. For sending native Sei tokens, use sendNative instead.
- /// Sends non-native tokens from one address to another.
- ///
- @param
- fromAddress The Sei address to send funds from.
- ///
- @param
- toAddress The Sei address to send funds to.
- ///
- @param
- denom The denomination of funds to send.
- ///
- @param
- amount The amount of the above denom to send.
- ///
- @return
- Whether the send was successfully executed.
- function
- send
- (
- address
- fromAddress
- ,
- address
- toAddress
- ,
- string
- memory
- denom
- ,
- uint256
- amount
-)
- external
- returns
- (
- bool
- success);
- sendNative
- : Enables a user to send native Sei to another user.
- /// Sends Sei to the specified user..
- ///
- @param
- toNativeAddress The Sei address of the recipient.
- ///
- @return
- Whether the tokens were successfully sent.
- function
- sendNative
- (
- string
- memory
- toNativeAddress
- ,
-)

- payable
- external
- returns
- (
- bool
- success);

Queries

- balance
- : Queries the balance of the given account for specified denom. /// Queries the balance of the given account for the specified denom.
- ///
- @param
- acc The Sei address of the account to query.
- ///
- @param
- denom The denomination to query for.
- ///
- @return
- The amount of denom held by acc.
- function
- balance
- (
- address
- acc
- ,
- string
- memory
- denom
-)
- external
- view
- returns
- (
- uint256
- amount);
- all_balances
- : Queries all balances of the given account.

/// Queries the balance of the given account for all balances. /// @param acc The Sei address of the account to query. /// @return balances for all coins/denoms. function

all_balances (address acc) external

view

returns (Coin [] memory response); * name * : Queries the name of the given denom. * /// Queries the name of the specified denom. * /// * @param * denom The denomination to query about. * /// * @return * The name of the specified denom. * function * name * (* string * memory * denom *) * external * view * returns * (* string * memory * response); * symbol * : Queries the symbol of the given denom. * /// Queries the symbol of the specified denom. * /// * @param * denom The denomination to query about. * /// * @return * The symbol of the specified denom. * function * symbol * (* string * memory * denom *) * external * view * returns * (* string * memory * response); * decimals * : Queries the number of decimal places for the given denom. * /// Queries the number of decimal places for the specified denom. * /// * @param * denom The denomination to query about. * /// * @return * The number of decimals for the specified denom. * function * decimals * (* string * memory * denom *) * external * view * returns * (* uint8 * response); * supply * : Queries the total supply of the given denom. * /// Queries the total supply of the specified denom. * /// * @param * denom The denomination to query about. * /// * @return * The total supply of the specified denom. * function * supply * (* string * memory * denom *) * external * view * returns * (* uint256 * response);

View the Bank precompile source code and the contract AB[here\(opens in a new tab\)](#) . Last updated on May 24, 2024
[Addresses Staking](#)