Send Native Coin

```

Copy usingSecretNET; usingSecretNET.Common; usingSecretNET.Common.Storage; usingSecretNET.Tx;

// Select a storage provider for the wallet // Docs: https://github.com/0xxCodemonkey/SecretNET#creating--initializing-the-wallet varstorageProvider=newAesEncryptedFileStorage("","SuperSecurePassword"); varcreateWalletOptions=newCreateWalletOptions(storageProvider);

// Import wallet from mnemonic phrase // Use key created snippet "Create a new Wallet" Walletwallet=null; if(awaitstorageProvider.HasPrivateKey()) { varstoredMnemonic=awaitstorageProvider.GetFirstMnemonic(); Console.WriteLine("Use stored mnemonic: "+storedMnemonic); wallet=awaitWallet.Create(storedMnemonic,options:createWalletOptions); Console.WriteLine("wallet.Address: "+wallet.Address); }

// get infos from https://docs.scrt.network/secret-network-documentation/development/connecting-to-the-network vargprcUrl="https://grpc.testnet.secretsaturn.net"; varchainId="pulsar-3";

// Create a connection to Secret Network node // Pass in a wallet that can sign transactions // Docs: https://github.com/0xxCodemonkey/SecretNET#creating--initializing-the-wallet varsecretClient=newSecretNetworkClient(gprcUrl,chainId,wallet:wallet);

vartx=awaitsecretClient.Tx.Bank.Send( toAddress:"secret1j8u7n4v93kjyqa7wzzrgjule8gh4adde36mnwd", amount:1000000, denom:"uscrt", txOptions:newTxOptions() { GasLimit=2_000_000} );

Console.WriteLine("Transaction: ",tx);

```