

Send Native Coin

```

```
Copy using SecretNET; using SecretNET.Common; using SecretNET.Common.Storage; using SecretNET.Tx;

// Select a storage provider for the wallet // Docs: https://github.com/0xxCodemonkey/SecretNET#creating--initializing-the-wallet
var storageProvider = new AesEncryptedFileStorage("", "SuperSecurePassword");
var createWalletOptions = new CreateWalletOptions(storageProvider);

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

// get infos from https://docs.scr.t.network/secret-network-documentation/development/connecting-to-the-network
var gprcUrl = "https://grpc.testnet.secretsaturn.net"; var chainId = "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
var secretClient = new SecretNetworkClient(gprcUrl, chainId, wallet: wallet);

var tx = await secretClient.Tx.Bank.Send(toAddress: "secret1j8u7n4v93kjq7wzzrgjule8gh4adde36mnwd", amount: 1000000,
denom: "uscrt", txOptions: new TxOptions() { GasLimit = 2_000_000 });

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

``` [Previous Deploying a Contract](#) [Next Query a Contract](#) Last updated 1 year ago