SNIP20 (SCRT Tokens)

Learn how to send SNIP20 tokens with SecretJS.

Secret Network Client Setup

```
Copy import{ SecretNetworkClient,Wallet }from"secretjs";

constwallet=newWallet("Your mnemonic words go here");

constsecretjs=newSecretNetworkClient({ chainId:"pulsar-3", url:"https://api.pulsar3.scrttestnet.com", wallet:wallet,
walletAddress:wallet.address, });
```

Transfer SNIP-20 tokens

```
Copy consttxExec=awaitsecretjs.tx.snip20.transfer( { sender:secretjs.address, contract_address, msg:{ transfer:{
recipient:accounts[1].address,amount:"2"} }, }, { gasLimit:5_000_000, }, );
```

Send SNIP-20 Tokens

```
Copy consttxExec=awaitsecretjs.tx.snip20.send( { sender:secretjs.address, contract_address, msg:{ send:{
recipient:accounts[1].address,amount:"2"} }, }, { gasLimit:5_000_000, }, );
```

Set Viewing Key and Query Balance

```
Copy consttxExec=awaitsecretjs.tx.snip20.setViewingKey({ sender:secretjs.address, contract_address:contractAddress,
msg:{ set_viewing_key:{ key:"hello"} }, });

consttxQuery=awaitsecretjs.query.snip20.getBalance({ address:secretjs.address, contract:{
address:contractAddress,codeHash:codeHash }, auth:{ key:"hello"}, });
```

Query Token Parameters

```
Copy consttxQuery=awaitsecretjs.query.snip20.getSnip20Params({ contract:{
address:contractAddress,codeHash:codeHash }, });

expect(txQuery).toEqual({ token_info:{ decimals:6, name:"Secret SCRT", symbol:"SSCRT", total_supply:"2", }, });
```

Get Transaction History

```
Copy consttxQuery=awaitsecretjs.query.snip20.getTransactionHistory({ contract:{
address:contractAddress,codeHash:codeHash }, address:secretjs.address, auth:{ key:"hello"}, page_size:10, });
```

Was this helpful? Edit on GitHub Export as PDF