

Using auto-wrapping on SNIP-20 tokens on mainnet with secret.js

For auto-wrapping public IBC tokens into their SNIP-20 tokens, SCRT Labs has created a auto-wrapping contract deployed on mainnet that everyone can use.

The contract info are:

Contract Address:secret198lmmh2fpj3weqhjcptkzl9pxygs23yn6dsev

Contract Hash:a6c421e3a60cf0e17931945b60d229eb4b4fce5f452122f34412cb937375ee27

To make use of the auto-wrapping contract, follow the guide below:

Initializing the Secret Network Client

We begin by initializing a new instance of `SecretNetworkClient` but with the caveat that this client is doing the transaction on the other IBC chain (e.g. Osmosis, Kujira etc.). This client is configured with several parameters, including the network URL (`chainLCD`), a wallet object, the wallet address, and the IBC chain's specific `chainId`:

...

```
const sourceOfflineSigner = (window as any).wallet.getOfflineSignerOnlyAmino(chainId)
const accounts = await sourceOfflineSigner.getAccounts()
```

```
const otherIBCChainSecretjs = new SecretNetworkClient({ url: chainLCD, chainId: chainId, wallet: sourceOfflineSigner,
walletAddress: accounts[0].address })
```

...

Creating the Transaction

The main action is in the transaction creation, specifically with the `otherIBCChainSecretjs.tx.ibc.transfer` function. This function is responsible for initiating an IBC transfer from the other IBC chain to Secret Network. Here's a breakdown of the key parameters:

- `sender`
 - : The address sending the tokens (in this case, the wallet address associated with `secretjs`)
- `client`.
- `receiver`
 - : The address of the wrap deposit contract on the destination chain.
- `source_channel`
 - : The IBC channel ID on the other Chain.
- `source_port`
 - : Set to "transfer"
- `type`, indicating the type of IBC transfer.
- `token`
 - : The amount and type of token to be transferred, here `123usct`
- `.`
- `timeout_timestamp`
 - : A UNIX timestamp indicating when the transaction should expire if not processed.
- `memo`
 - : A JSON object containing instructions for the wrap deposit contract.
- `.`

...

```
const wrapDepositContractAddress = "secret198lmmh2fpj3weqhjcptkzl9pxygs23yn6dsev"
const ibcChannelIdOnChain2 = "channel-XX"
```

```
const tokenContractAddress = "secret1....."
const tokenCodeHash = "somecontractcodehash"
```

```
let tx = await otherIBCChainSecretjs.tx.ibc.transfer({ sender: otherIBCChainSecretjs.address, // The address on the other IBC
chain receiver: wrapDepositContractAddress, source_channel: ibcChannelIdOnChain2, source_port: "transfer", token: {
amount: "123", denom: "uXXX" // the token you want to IBC in and autowrap },
timeout_timestamp: String(Math.floor(Date.now() / 1000) + 10 * 60), // 10 minutes memo: JSON.stringify({ wasm: {
contract: wrapDepositContractAddress, msg: { wrap_deposit: { snip20_address: tokenContractAddress,
snip20_code_hash: tokenCodeHash, recipient_address: secretAddress, // the address on Secret Network }, }, }, }, {
broadcastCheckIntervalMs: 100, gasLimit: 100_000, ibcTxOptions: { resolveResponsesCheckIntervalMs: 250, }, }, );
```

...

The Auto-Wrapping Mechanism

The auto-wrapping feature comes into play in the memo field of the transaction. This field carries a JSON object that instructs the wrap deposit contract on how to handle the incoming tokens. The instructions include:

- `wrap_deposit`
- : Indicates the action to be taken is a deposit wrap.
- `snip20_address`
- `andsnip20_code_hash`
- : Details of the SNIP-20 token on Secret Network. See [Secret \(SNIP 20\) token contracts list](#)
- for the exact details for each token.
- `recipient_address`
- : The address on the Secret Network where the wrapped token should be sent to.
-

This auto-wrapping mechanism is a powerful feature of the Secret Network. When the SNIP-20 tokens are transferred via IBC to another chain (like the Secret Network), they are automatically wrapped into a Secret Token (a privacy-preserving token on the Secret Network). This wrapping is done by the wrap deposit contract specified in the memo field.

Transaction Broadcasting and Options

Finally, the transaction is broadcasted with specific options like `broadcastCheckIntervalMs` and `ibcTxOptions`. The `gasLimit` setting is the maximum amount of gas that can be consumed by the transaction.

In summary, this code snippet demonstrates a sophisticated use of the Secret Network's IBC hooks to automatically wrap public IBC tokens into their SNIP-20 tokens. This process enhances the interoperability and privacy features of the tokens when they are transferred across different blockchains.

Last updated 3 months ago On this page * [Initializing the Secret Network Client](#) * [Creating the Transaction](#) * [The Auto-Wrapping Mechanism](#) * [Transaction Broadcasting and Options](#)

Was this helpful? [Edit on GitHub](#) [Export as PDF](#)