

---

# **web3-fusion-extend Documentation**

**Fusion.org**

**Feb 15, 2019**



<b>1</b>	<b>Contributing</b>	<b>3</b>
1.1	Submitting a pull request . . . . .	3
1.2	Getting started . . . . .	4
1.3	License . . . . .	5
<b>2</b>	<b>Getting Started</b>	<b>7</b>
2.1	Installation . . . . .	7
2.2	Usage . . . . .	8
2.3	Contribute! . . . . .	8
2.4	Requirements . . . . .	8
2.5	Testing (mocha) . . . . .	9
2.6	Community . . . . .	9
2.7	Documentation . . . . .	9
2.8	License . . . . .	9
<b>3</b>	<b>autoPurchaseTickets</b>	<b>11</b>
3.1	Ticket purchase applicaton . . . . .	11
<b>4</b>	<b>blockexplorerapi</b>	<b>13</b>
4.1	Overview . . . . .	13
4.2	Fusion Org and its public explorer api . . . . .	13
4.3	Installation . . . . .	14
4.4	API Commands . . . . .	14
<b>5</b>	<b>FSNConstants</b>	<b>17</b>
5.1	Constants . . . . .	17
<b>6</b>	<b>fsn</b>	<b>21</b>
6.1	allAssets . . . . .	21
6.2	allNotation . . . . .	22
6.3	allSwaps . . . . .	23
6.4	allTickets . . . . .	23
6.5	allTicketsByAddress . . . . .	24
6.6	totalNumberOfTickets . . . . .	24
6.7	totalNumberOfTicketsByAddress . . . . .	25
6.8	assetToTimeLock . . . . .	25
6.9	getHexDate . . . . .	26

6.10	timeLockToAsset	26
6.11	timeLockToTimeLock	27
6.12	buyTicket	28
6.13	genAsset	28
6.14	genNotation	29
6.15	sendAsset	29
6.16	decAsset	30
6.17	incAsset	30
6.18	getAddressByNotation	31
6.19	getAllBalances	31
6.20	getAllTimeLockBalances	32
6.21	getAsset	32
6.22	getBalance	33
6.23	getNotation	33
6.24	getTimeLockBalance	34
6.25	makeSwap	34
6.26	recallSwap	35
6.27	takeSwap	35
<b>7</b>	<b>fsntx</b>	<b>37</b>
7.1	sendRawTransaction	37
7.2	buildGenNotationTx	38
7.3	genNotation	38
7.4	buildGenAssetTx	39
7.5	genAsset	39
7.6	buildSendAssetTx	40
7.7	sendAsset	40
7.8	buildAssetToTimeLockTx	41
7.9	assetToTimeLock	41
7.10	buildTimeLockToTimeLockTx	42
7.11	timeLockToTimeLock	43
7.12	buildTimeLockToAssetTx	43
7.13	timeLockToAsset	44
7.14	buildBuyTicketTx	45
7.15	buyTicket	45
7.16	buildIncAssetTx	46
7.17	incAsset	46
7.18	buildDecAssetTx	47
7.19	decAsset	47
7.20	buildMakeSwapTx	48
7.21	makeSwap	48
7.22	buildRecallSwapTx	49
7.23	recallSwap	49
7.24	buildTakeSwapTx	50
7.25	takeSwap	50

Fusion is an enterprise financial infrastructure blockchain solution simplifying the full digital life of any asset.

With the ethereum world users are familiar that they have a wallet and a balance. If user own ERC20 or ERC721 tokens wallets applications hide the technicalities of calling contracts to get their values.

Fusion evolutionary approach moves the asset and its balance to the users ethereum wallet.

No contracts are needed as multiple balances each with a unique asset Id can be contained on the chain in the wallet.

Other inflationary practices to code such as creating escrows and atomic swaps are also eliminated by simply having the asset in the users account.

Fusion allows all assets to have time periods, these time periods enable safe lending of an asset.

So a balance of 1 Fusion Token can be Timelocked for 30 days, sent to another account.

At the end of the 30 days that other account no longer can access that value in their wallet.

All done natively on the chain without contracts.

The goal of Fusion is to:

- **make assets (tokens) a natural part of send and receive of value**
  - eliminate need for ERC20 contracts
  - eliminate need for ERC721 contracts
  - simply the familiar paradigm of from, to, value -> from, to, assetId, value
- **introduce the concept of timelock of an asset as a natural wallet feature**
  - split the use of any asset into any sliver of time
  - allow that time portion of that asset to be securely lent to someone
- **quantum swap - introduce a one transaction method to exchange assets between parties**
  - the ability to exchange assets between accounts in one transaction on the blockchain without chance of error
  - assets that are timelocked can be exchanged
  - swaps can be made public to all users or a targeted user.

The purpose of the web3-fusion-extend library is to take the popular web3.js package and open it up to the extended functionality of that the Fusion blockchain server provides.

Contents:

[Keyword Index](#), [Search Page](#)

---

**Note:** This documentation is work in progress and web3-fusion-extend 1.0 stable is not yet released! You can find the current documentation for web3-fusion-extend 0.x.x at.

---



If you're reading this, you're awesome! Thank you for helping us make this project great and being a part of the web3-fusion-extend community. Here are a few guidelines that will help you along the way.

## 1.1 Submitting a pull request

web3-fusion-extend is a community project, so pull requests are always welcome, but, before working on a large change, it is best to open an issue first to discuss it with the maintainers.

When in doubt, keep your pull requests small. To give a PR the best chance of getting accepted, don't bundle more than one feature or bug fix per pull request. It's always best to create two smaller PRs than one big one.

As with issues, please make sure your title clearly explains the issue.

When adding new features or modifying existing, please attempt to include tests to confirm the new behaviour. You can read more about our test setup [here](#).

### 1.1.1 Branch Structure

All stable releases are tagged ([view tags](#)). At any given time, `master` represents the latest development version of the library. Patches or hotfix releases are prepared on an independent branch.

`master` is unsafe

We will do our best to keep `master` in good shape, with tests passing at all times. However, in order to move fast, we will make API changes that your application might not be compatible with.

### 1.1.2 How to increase the chance of being accepted?

We will only accept a pull request for which all tests pass. Make sure the following is true:

- The branch is not behind `master`.

- **If a feature is being added:**
  - If the result was already achievable with the core library, explain why this feature needs to be added to the core.
  - It includes relevant tests.
  - If this is a common use case, considered adding an example to the documentation.
- If a bug is being fixed, test cases that fail without the fix are included.
- The code is formatted (run `yarn prettier`).
- The code is linted (run `yarn lint`).
- If API documentation is being changed in the source, `yarn docs:api` was run.
- If prop types were changed, the TypeScript declarations were updated.
- If TypeScript declarations were changed, `yarn typescript` passed.
- The PR title follows the pattern `[Component] Imperative commit message`. (See: [\(How to Write a Git Commit Message\)](#) for a great explanation)

## 1.2 Getting started

Please create a new branch from an up to date master on your fork. (Note, urgent hotfixes should be branched off the latest stable release rather than master)

1. Fork the web3-fusion-extend repository on Github
2. Clone your fork to your local machine `git clone git@github.com:<yourname>/web3-fusion-extend.git`
3. Create a branch `git checkout -b my-topic-branch`
4. Make your changes, lint, then push to to GitHub with `git push --set-upstream origin my-topic-branch`.
5. Visit GitHub and make your pull request.

If you have an existing local repository, please update it before you start, to minimise the chance of merge conflicts.

```
git remote add upstream git@github.com:FusionFoundation/web3-fusion-extend.git
git checkout master
git pull upstream master
git checkout -b my-topic-branch
yarn
```

### 1.2.1 Coding style

Please follow the coding style of the project. web3-fusion-extend uses eslint, so if possible, enable linting in your editor to get real-time feedback. The linting rules can be run manually with the following command `yarn lint`.

You can also run `yarn prettier` to reformat the code.

Finally, when you submit a pull request, they are run again by Circle CI, but hopefully by then your code is already clean!



## 1.3 License

By contributing your code to the FusionFoundation/web3-fusion-extend GitHub repository, you agree to license your contribution under the MIT license.

---

**Note:** This documentation is work in progress and web3-fusion-extend 1.0 stable is not yet released! You can find the current documentation for web3-fusion-extend 0.x.x at.

---



web3-fusion-extend is a collection of libraries which allow you to interact with a local or remote fusion node, using a HTTP or IPC connection.

Fusion offers a radical approach to representing value within a block chain environment.

A public address contain multiple assets and balances for these assets.

An assetId is the id returned when an asset is created and actions can be performed on it.

The asset creator also has the ability to increase and decrease supply.

This enables cross chain and cross functional systems to be built that enable the interchange of assets.

Assets can also be TimeLocked. When an asset is time locked its ownership is leant for the period specified. At the end of time lock period the rights of the asset are returned to the original owner.

With the representation of assets, the need to exchange assets securely and simply becomes paramount.

**The Fusion protocol introduces quantumSwaps which are composed of three functions:** makeSwap - tell others what you will exchange for your asset recallSwap - cancel the request for an exchange takeSwap - exchange your asset for the other parties asset listed in the make swap

This package extends the Ethereum compatible JavaScript API which implements the [Generic JSON RPC](#) spec to support the Fusion protocol.

It's available on npm as a node module. You need to run a local Ethereum node to use this library.

- [Documentation](#)

## 2.1 Installation

Node.js

```
npm install web3-fusion-extend
```

Yarn

```
yarn add web3-fusion-extend
```

## 2.2 Usage

Create a web3 object as your normally would and then call web3FusionExtend with that object. web3 will then have two additional interfaces (fsn and fsntx)

```
var web3FusionExtend = require('web3-fusion-extend')
web3 = new Web3(provider);
web3 = web3FusionExtend.extend(web3)
console.log(web3.fsn.consts.FSNToken);
```

```
console.log(web3); // {fsn: .., fsntx: ...} // It's here!
```

There you go, now you can use it:

```
var balance = web3.eth.getBalance(coinbase);
web3.fsn
  .getAllBalances( web3.eth.coinbase ) // fsn supports multiple assets and balances
  ↪on an address
  .then( balances => {
    console.log( balances )
    assert( balances[web3.fsn.consts.FSNToken] , "there should be a balance for
  ↪fusion tokens always" )
    done();
  })
  .catch(err => {
    done(err);
  });
```

You can find more examples in the [test](#) directory.

There is also a full block explorer api written as an example.

- [BlockExplorerApi](#)

## 2.3 Contribute!

We'd greatly appreciate any *contribution* you make.

## 2.4 Requirements

Node.js npm

```
# On Linux:
sudo apt-get update
sudo apt-get install nodejs
sudo apt-get install npm
sudo apt-get install nodejs-legacy
```

## 2.5 Testing (mocha)

When testing a connect string to a local fusion node and a wallet address is needed as environment variables

```
CONNECT_STRING="ws://3.16.110.25:9001" WALLET_ADDRESS=  
↪"0x4A5a7Aa4130e407d3708dE56db9000F059200C62" npm test
```

## 2.6 Community

- [Gitter](#)
- [Forum](#)

## 2.7 Documentation

Documentation

- [Overview](#)
- [fsn](#)
- [fsntx](#)
- [Constants](#)

## 2.8 License

LGPL-3.0+ © 2015 Contributors



---

## autoPurchaseTickets

---

### 3.1 Ticket purchase applicaton

An application to keep tickets at a constant level built with the web3-fusion-extend package

- autoPurchaseTicket.js

#### 3.1.1 Example

```
node autoPurchaseTicket --c "wss://example.com" -p "./password.txt" -k "./keystore.key  
↪" -n 10
```

```
-c --connectString web socket gateway to connect to  
-k --keyStore keystore file to use  
-p --passPharseFile key file  
-g -- gasPrice gas price 1 - 100 (defaults to 2 gwei)  
-n --Number of tickets to purchase
```





### 4.1 Overview

This block explorer api section shows how to collect all information from the fusion block chain and update a mysql database.

You will need a node for the application readAllBlocksToDatabase.js to communicate with as well as a mysql database to store the information.

You will need an environment string called DB\_CONNECT\_STRING

It can be passed on the command line.

```
DB_CONNECT_STRING="{ 'host': 'mysqlserver', 'user': 'adminuser', 'password': 'password',  
→ 'database': 'fusionblockdb', 'connectionLimit': 100 }" node readAllBlocksToDatabase.js
```

You can then host the api server for the database via:

```
DB_CONNECT_STRING="{ 'host': 'mysqlserver', 'user': 'adminuser', 'password': 'password',  
→ 'database': 'fusionblockdb', 'connectionLimit': 100 }" nodemon node ./bin/www
```

### 4.2 Fusion Org and its public explorer api

Fusion organization keeps an api endpoint open at <https://api.fusionnetwork.io> to assist in application development

You can try commands like <https://api.fusionnetwork.io/blocks/latest> or <https://api.fusionnetwork.io/transactions/latest>

Note if you are not a developer the results may look scary but they are the actual last block or transaction info

## 4.3 Installation

```
npm install
```

## 4.4 API Commands

If running locally replace `api.fusionnetwork.io` with your own server link

### 4.4.1 Assets

- <http://api.fusionnetwork.io/assets/0xbbd28ab973a7be78af3d8a3c3f1097c87fc020b2bd9270aa292518e8a93c32ae>
- <http://api.fusionnetwork.io/assets/all?page=0&size=2&sort=desc>

### 4.4.2 Balance

- <http://api.fusionnetwork.io/balances/0xC4A9441afB052cB454240136CCe71Fb09316EA94>
- <http://api.fusionnetwork.io/balances/all?page=0&size=2&sort=asc>

### 4.4.3 Blocks

- <http://api.fusionnetwork.io/blocks/latest>
- <http://api.fusionnetwork.io/blocks/300>
- <http://api.fusionnetwork.io/blocks/all?sort=asc&page=2&size=10&field=height&sort=desc>
- <http://api.fusionnetwork.io/blocks/range?to=10&from=100>

fields can be: [ timestamp, hash , numberOfTransactions, height ]

### 4.4.4 Swaps

- <http://api.fusionnetwork.io/swaps/0xbbd28ab973a7be78af3d8a3c3f1097c87fc020b2bd9270aa292518e8a93c32ae>
- <http://api.fusionnetwork.io/swaps/all?page=0&size=2&sort=asc>

### 4.4.5 Transactions

- <http://api.fusionnetwork.io/transactions/latest>
- <http://api.fusionnetwork.io/transactions/0x346aab726aa05808698ec9aba5da4e4c4574863e87951b5107d3fdabc290bbaa>
- <http://api.fusionnetwork.io/transactions/all?sort=asc&page=2&size=10&field=height>

fields can be: [ timestamp, hash , type, block , asset ]

Return an array of transactions from a - seperated array - <http://api.fusionnetwork.io/transactions/ts?ts=address1-address2>

## 4.4.6 Fusion Price

- **last price**
  - <http://api.fusionnetwork.io/fsnprice>
- **historical prices**
  - <http://api.fusionnetwork.io/fsnprice/?page=0&size=2&sort=asc>
- **last two prices**
  - <http://api.fusionnetwork.io/fsnprice/?page=0&size=2&sort=desc>

## 4.4.7 Search

- <http://api.fusionnetwork.io/search/{}block,blockhash,transaction,address{}>
- miner leaderboard <http://api.fusionnetwork.io/leaderboard>
- **Ticket purchase applicaton**
  - `autoPurchaseTicket.js`

Example: `node autoPurchaseTicket -c"wss://example.com" -p "./password.txt" -k "./keystore.key" -n 10`

```
-c --connectString web socket gateway to connect to
-k --keyStore keystore file to use
-p --passPharseFile key file
-n --Number of tickets to purchase
```



- *Overview*
- *fsn*
- *fsntx*

### 5.1 Constants

```
web3.fsn.consts = {}
```

#### 5.1.1 FSNToken

```
FSNToken: "0xffffffffffffffffffffffffffffffffffffffff"
```

---

#### 5.1.2 TicketLogAddress

```
TicketLogAddress: "0xffffffffffffffffffffffffffffffffffffe",
```

---

#### 5.1.3 TicketLogAddress\_Topic\_To\_Function

```
TicketLogAddress_Topic_To_Function: {  
  0: "ticketSelected",  
  1: "ticketReturn",  
  2: "ticketExpired"  
},
```

---

### 5.1.4 TicketLogAddress\_Topic\_ticketSelected

```
TicketLogAddress_Topic_ticketSelected:  
"0x0000000000000000000000000000000000000000000000000000000000000000",
```

### 5.1.5 TicketLogAddress\_Topic\_ticketReturn

```
TicketLogAddress_Topic_ticketReturn:  
"0x0000000000000000000000000000000000000000000000000000000000000001",
```

### 5.1.6 TicketLogAddress\_Topic\_ticketExpired

```
TicketLogAddress_Topic_ticketExpired:  
"0x0000000000000000000000000000000000000000000000000000000000000002",
```

### 5.1.7 FSNCallAddress

```
FSNCallAddress: "0xffffffffffffffffffffffffffffffffffffffff",
```

### 5.1.8 FSNCallAddress\_Topic\_To\_Function

```
FSNCallAddress_Topic_To_Function: {  
  // GenNotationFunc wacom  
  0: "GenNotationFunc", // = iota  
  // GenAssetFunc wacom  
  1: "GenAssetFunc",  
  // SendAssetFunc wacom  
  2: "SendAssetFunc",  
  // TimeLockFunc wacom  
  3: "TimeLockFunc",  
  // BuyTicketFunc wacom  
  4: "BuyTicketFunc",  
  // AssetValueChangeFunc wacom  
  5: "AssetValueChangeFunc",  
  // MakeSwapFunc wacom  
  6: "MakeSwapFunc",  
  // RecallSwapFunc wacom  
  7: "RecallSwapFunc",  
}
```

(continues on next page)

(continued from previous page)

```
// TakeSwapFunc wacom
8: "TakeSwapFunc"
  // MakeSwapFuncExt wacom
  9: "MakeSwapFuncExtOld",
  // MakeSwapFuncExt wacom
  10: "MakeSwapFuncExt",
  // TakeSwapFuncExt wacom
  11: "TakeSwapFuncExt",
  // AssetValueChangeFunc wacom
  12: "AssetValueChangeExtFunc"
},
```

### 5.1.9 FSNCallAddress\_Topic\_GenNotationFunc

```
FSNCallAddress_Topic_GenNotationFunc:
  "0x0000000000000000000000000000000000000000000000000000000000000000",
```

### 5.1.10 FSNCallAddress\_Topic\_GenAssetFunc

```
FSNCallAddress_Topic_GenAssetFunc:
  "0x0000000000000000000000000000000000000000000000000000000000000001",
```

### 5.1.11 FSNCallAddress\_Topic\_SendAssetFunc

```
FSNCallAddress_Topic_SendAssetFunc:
  "0x0000000000000000000000000000000000000000000000000000000000000002",
```

### 5.1.12 FSNCallAddress\_Topic\_TimeLockFunc

```
FSNCallAddress_Topic_TimeLockFunc:
  "0x0000000000000000000000000000000000000000000000000000000000000003",
```

### 5.1.13 FSNCallAddress\_Topic\_BuyTicketFunc

```
FSNCallAddress_Topic_BuyTicketFunc:
  "0x0000000000000000000000000000000000000000000000000000000000000004",
```

### 5.1.14 FSNCallAddress\_Topic\_AssetValueChangeFunc

```
FSNCallAddress_Topic_AssetValueChangeFunc:  
"0x0000000000000000000000000000000000000000000000000000000000000005",
```

---

### 5.1.15 FSNCallAddress\_Topic\_MakeSwapFunc

```
FSNCallAddress_Topic_MakeSwapFunc:  
"0x0000000000000000000000000000000000000000000000000000000000000006",
```

---

### 5.1.16 FSNCallAddress\_Topic\_RecallSwapFunc

```
FSNCallAddress_Topic_RecallSwapFunc:  
"0x0000000000000000000000000000000000000000000000000000000000000007",
```

---

### 5.1.17 FSNCallAddress\_Topic\_TakeSwapFunc

```
FSNCallAddress_Topic_TakeSwapFunc:  
"0x0000000000000000000000000000000000000000000000000000000000000008"
```

---

**FSNCallAddress\_Topic\_MakeSwapFuncExtOld:** "0x00"

**FSNCallAddress\_Topic\_MakeSwapFuncExt:** "0x00"

**FSNCallAddress\_Topic\_TakeSwapFuncExt:** "0x00"

**FSNCallAddress\_Topic\_AssetValueChangeExtFunc:** "0x00"

---

**Note:** This documentation is work in progress and web3-fusion-extend 1.0 stable is not yet released! You can find the current documentation for web3-fusion-extend 0.x.x at.

---



## FSN documentation

- *Overview*
- *fsn*
- *fsntx*
- *Constants*

```
fsn.assetToTimeLock({asset :  
↪ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.  
↪ coinbase, to: "0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165", start: "0x100", end: "0x200",  
↪ value: "0x100"}, "123456")
```

```
fsn.assetToTimeLock({asset :  
↪ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.  
↪ coinbase, to: "0x4A5a7Aa4130e407d3708dE56db9000F059200C62", start: "0x100", end: "0x200",  
↪ value: "0x100"}, "123456")
```

common params CP from send from or call member gas gas limit gasPrice gas price nonce send from or call member account nonce

common send params CSP CP asset the asset ID to receiver value amount

## 6.1 allAssets

```
fsn.allAssets()  
...
```

allAssets get the all assets list no params

### 6.1.1 Parameters

none

### 6.1.2 Returns

Asset Object: With the following methods:

- ID Hash: Description
- Owner Address: Description
- Name string: Description
- Symbol string: Description
- Decimals uint8: Description
- Total \*big.Int `json:",string"`: Description
- CanChange bool: Description

### 6.1.3 Example

```
fsn.allAssets()
```

## 6.2 allNotation

```
fsn.allNotation()  
...
```

allNotation get the all notation no params

### 6.2.1 Parameters

none

### 6.2.2 Returns

Object: With the following methods:

- Object: Description

### 6.2.3 Example

```
fsn.allNotation()
```

## 6.3 allSwaps

```
fsn.allSwaps()  
...
```

allSwaps get the all quantum swap list no params

### 6.3.1 Parameters

none

### 6.3.2 Returns

Object: With the following methods:

- Object: Description

### 6.3.3 Example

```
fsn.allSwaps()
```

## 6.4 allTickets

```
fsn.allTickets()  
...
```

allTickets get the all notation no params

### 6.4.1 Parameters

none

### 6.4.2 Returns

Object: With the following methods:

- Object: Description

### 6.4.3 Example

```
fsn.allTickets()
```

## 6.5 allTicketsByAddress

```
fsn.allTicketsByAddress(eth.coinbase)
...
```

allTicketsByAddress get all tickets by address address the user's address

### 6.5.1 Parameters

none

### 6.5.2 Returns

Object: With the following methods:

- Object: Description

### 6.5.3 Example

```
fsn.allTicketsByAddress(eth.coinbase)
```

## 6.6 totalNumberOfTickets

```
fsn.totalNumberOfTickets()
...
```

totalNumberOfTickets return number of active tickets no params

### 6.6.1 Parameters

none

### 6.6.2 Returns

Object: With the following methods:

- Object: Description

### 6.6.3 Example

```
fsn.totalNumberOfTickets()
```

## 6.7 totalNumberOfTicketsByAddress

```
fsn.totalNumberOfTicketsByAddress (eth.coinbase)
...
```

totalNumberOfTicketsByAddress return number of tickets an address controls address address that bought tickets

### 6.7.1 Parameters

none

### 6.7.2 Returns

Object: With the following methods:

- Object: Description

### 6.7.3 Example

```
fsn.totalNumberOfTicketsByAddress (eth.coinbase)
```

## 6.8 assetToTimeLock

```
fsn.assetToTimeLock ({asset :
↳ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.
↳ coinbase, to: "0xa7455DF112c953F3c73c2C25559965e1A8a20024", start: "0x1", end: "0x2A300",
↳ value: "0x1400000000000000"}, "123456")
...
```

assetToTimeLock send the asset to time lock CSP see the top startTime the start time of the time lock endTime the end time of the time lock password the account password

### 6.8.1 Parameters

none

### 6.8.2 Returns

Object: With the following methods:

- Object: Description

### 6.8.3 Example

```
fns.assetToTimeLock({asset :
↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fns.
↳coinbase, to: "0xa7455DF112c953F3c73c2C25559965e1A8a20024", start: "0x1", end: "0x2A300",
↳value: "0x1400000000000000"}, "123456")

fns.assetToTimeLock({asset :
↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fns.
↳coinbase, to: "0xa7455DF112c953F3c73c2C25559965e1A8a20024", start: "0x2a900", end:
↳"0x3A300", value: "0x1400000000000000"}, "123456")

fns.assetToTimeLock({asset :
↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fns.
↳coinbase, to: "0xa7455DF112c953F3c73c2C25559965e1A8a20024", start: getHexDate('2018-12-
↳01') , end: getHexDate('2019-01-01'), value: "0x1340000000000000"}, "123456")
```

## 6.9 getHexDate

```
fns.getHexDate('2018-12-01')
...
```

getHexDate helper function to convert date to posix time in hex

### 6.9.1 Parameters

none

### 6.9.2 Returns

Object: With the following methods:

- Object: Description

### 6.9.3 Example

```
fns.getHexDate('2018-12-01')
fns.assetToTimeLock({asset :
↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fns.
↳coinbase, to: fns.coinbase, start: "0x1", end: "0x10000", value: "0x100"}, "123456")
```

## 6.10 timeLockToAsset

```
fns.timeLockToAsset({asset :
↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fns.
↳coinbase, to: fns.coinbase, start: "0x0", end: "0x0", value: "0x100"}, "123456")
...
```

timeLockToAsset send the time lock to asset CSP see the top startTime the start time of the time lock endTime the end time of the time lock password the account password

### 6.10.1 Parameters

none

### 6.10.2 Returns

Object: With the following methods:

- Object: Description

### 6.10.3 Example

```
fns.timeLockToAsset ({asset :
↳ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fns.
↳ coinbase, to: fns.coinbase, start: "0x0", end: "0x0", value: "0x100"}, "123456")
```

## 6.11 timeLockToTimeLock

```
fns.timeLockToTimeLock ({asset :
↳ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fns.
↳ coinbase, to: "0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165", start: "0x101", end: "0x200",
↳ value: "0x100"}, "123456")
...
```

timeLockToTimeLock send the time lock CSP see the top start`Time` the start time of the time lock end`Time` the end time of the time lock password the account password

### 6.11.1 Parameters

none

### 6.11.2 Returns

Object: With the following methods:

- Object: Description

### 6.11.3 Example

```
fns.timeLockToTimeLock ({asset :
↳ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fns.
↳ coinbase, to: "0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165", start: "0x101", end: "0x200",
↳ value: "0x100"}, "123456")
```

## 6.12 buyTicket

```
fsn.buyTicket({from:fsn.coinbase},"123456")  
...
```

buyTicket buy the ticket CP see the top and the “from” ignore from who buy the ticket password the account password

### 6.12.1 Parameters

none

### 6.12.2 Returns

Object: With the following methods:

- Object: Description

### 6.12.3 Example

```
fsn.buyTicket({from:fsn.coinbase},"123456")
```

## 6.13 genAsset

```
fsn.genAsset({from:fsn.coinbase,name:"FusionTest",symbol:"FST",decimals:1,total:"0x200  
↔"}, "123456")  
fsn.genAsset({from:"0x91db50f5c36ae7616009d4e94462dca4d4c7e833",name:"JONESY",symbol:  
↔"JSY",decimals:1,total:"0x2000000000"}, "123123123")  
...
```

genAsset generate a asset CP see the top and the “from” ignore from who gen the asset and the owner of the asset name the name of asset symbol the symbol of asset decimals the asset decimal digit total the total number of the asset and the owner will be get same number asset CanChange whether asset can be incremented or decremented by the owner [optional] password the account password

### 6.13.1 Parameters

none

### 6.13.2 Returns

Object: With the following methods:

- Object: Description



### 6.13.3 Example

```
fsn.genAsset({from:fsn.coinbase,name:"FusionTest",symbol:"FST",decimals:1,total:"0x200
↪"}, "123456")
fsn.genAsset({from:"0x91db50f5c36ae7616009d4e94462dca4d4c7e833",name:"JONESY",symbol:
↪"JSY",decimals:1,total:"0x2000000000"}, "123123123")
```

## 6.14 genNotation

```
fsn.genNotation({from:fsn.coinbase}, "123456")
...
```

genNotation gen a notation for a account CP see the top and the “from” ignore from who gen the notation password the account password

### 6.14.1 Parameters

none

### 6.14.2 Returns

Object: With the following methods:

- Object: Description

### 6.14.3 Example

```
fsn.genNotation({from:fsn.coinbase}, "123456")
```

## 6.15 sendAsset

```
fsn.sendAsset({from:fsn.coinbase,to:"0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165",
↪value:"0x1",asset:
↪"0x514a46f34e6eb0a98abb3595c4aec33ca8ddf69f135c8fed89e78d0808047965"}, "123456")

fsn.sendAsset({from:fsn.coinbase,to:"0x91db50f5c36ae7616009d4e94462DcA4D4c7e833",
↪value:"0x2",asset:
↪"0x88d18f81620e5684e880dddcf0b6c167a9154d4c499bc9fad47b98634110eeec"}, "123456")
...
```

sendAsset send asset to other account CSP see the top

### 6.15.1 Parameters

none

## 6.15.2 Returns

Object: With the following methods:

- Object: Description

## 6.15.3 Example

```
fnsn.sendAsset ({from:fnsn.coinbase,to:"0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165",
↪value:"0x1",asset:
↪"0x514a46f34e6eb0a98abb3595c4aec33ca8ddf69f135c8fed89e78d0808047965"}, "123456")

fnsn.sendAsset ({from:fnsn.coinbase,to:"0x91db50F5c36aE7616009d4e94462DcA4D4c7e833",
↪value:"0x2",asset:
↪"0x88d18f81620e5684e880dddcf0b6c167a9154d4c499bc9fad47b98634110eeec"}, "123456")
```

## 6.16 decAsset

```
fnsn.decAsset ({from:fnsn.coinbase,to:"0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165",value:
↪"0x1",asset:"0x514a46f34e6eb0a98abb3595c4aec33ca8ddf69f135c8fed89e78d0808047965"},
↪"123456")
...
```

decAsset decrease account asset balance CSP see the top and the “from”,”to” ignore from the asset owner to the dec account password the account password

### 6.16.1 Parameters

none

### 6.16.2 Returns

Object: With the following methods:

- Object: Description

### 6.16.3 Example

```
fnsn.decAsset ({from:fnsn.coinbase,to:"0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165",value:
↪"0x1",asset:"0x514a46f34e6eb0a98abb3595c4aec33ca8ddf69f135c8fed89e78d0808047965"},
↪"123456")
```

## 6.17 incAsset

```
fnsn.incAsset ({from:fnsn.coinbase,to:"0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165",value:
↪"0x1",asset:"0x514a46f34e6eb0a98abb3595c4aec33ca8ddf69f135c8fed89e78d0808047965"},
↪"123456")
...
```

incAsset increase account asset balance CSP see the top and the “from”,”to” ignore from the asset owner to the inc account password the account password

### 6.17.1 Parameters

none

### 6.17.2 Returns

Object: With the following methods:

- Object: Description

### 6.17.3 Example

```
fsn.incAsset ({from:fsn.coinbase,to:"0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165",value:
↪ "0x1",asset:"0x514a46f34e6eb0a98abb3595c4aec33ca8ddf69f135c8fed89e78d0808047965"},
↪ "123456")
```

## 6.18 getAddressByNotation

```
fsn.getAddressByNotation(104)
...
```

getAddressByNotation get the notation of the address notation account notation

### 6.18.1 Parameters

none

### 6.18.2 Returns

Object: With the following methods:

- Object: Description

### 6.18.3 Example

```
fsn.getAddressByNotation(104)
```

## 6.19 getAllBalances

```
fsn.getAllBalances("0x9c48c796cb0bed51a14291bc8cc56dab5aed7b5c")
...
```

getAllBalances get all assets balances address the user’s address blockNumber default now block number

### 6.19.1 Parameters

none

### 6.19.2 Returns

Object: With the following methods:

- Object: Description

### 6.19.3 Example

```
fsn.getAllBalances ("0x9c48c796cb0bed51a14291bc8cc56dab5aed7b5c")
```

## 6.20 getAllTimeLockBalances

```
fsn.getAllTimeLockBalances ("0x9c48c796cb0bed51a14291bc8cc56dab5aed7b5c")  
...
```

getAllTimeLockBalances get all time lock balances address the user's address blockNumber default now block number

### 6.20.1 Parameters

none

### 6.20.2 Returns

Object: With the following methods:

- Object: Description

### 6.20.3 Example

```
fsn.getAllTimeLockBalances ("0x9c48c796cb0bed51a14291bc8cc56dab5aed7b5c")
```

## 6.21 getAsset

```
fsn.getAsset ("0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff")  
...
```

getAsset get the asset info assetID the asset ID

### 6.21.1 Parameters

none

## 6.21.2 Returns

Object: With the following methods:

- Object: Description

## 6.21.3 Example

```
fsn.getAsset("0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff")
```

## 6.22 getBalance

```
fsn.getBalance("0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff",
  →"0x9c48c796cb0bed51a14291bc8cc56dab5aed7b5c"),
...

```

getBalance like name assetID the asset ID address the user's address blockNumber default now block number

### 6.22.1 Parameters

none

### 6.22.2 Returns

Object: With the following methods:

- Object: Description

### 6.22.3 Example

```
fsn.getBalance("0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff",
  →"0x9c48c796cb0bed51a14291bc8cc56dab5aed7b5c")
```

## 6.23 getNotation

```
fsn.getNotation("0x9c48c796cb0bed51a14291bc8cc56dab5aed7b5c")
...

```

getNotation get the notation of address address the account address

### 6.23.1 Parameters

none

### 6.23.2 Returns

Object: With the following methods:

- Object: Description

### 6.23.3 Example

```
fsn.getNotation("0x9c48c796cb0bed51a14291bc8cc56dab5aed7b5c")
```

## 6.24 getTimeLockBalance

```
fsn.getTimeLockBalance(  
  ↪ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff",  
  ↪ "0x9c48c796cb0bed51a14291bc8cc56ed7b5c")  
...
```

getTimeLockBalance like name assetID the asset ID address the user's address blockNumber default now block number

### 6.24.1 Parameters

none

### 6.24.2 Returns

Object: With the following methods:

- Object: Description

### 6.24.3 Example

```
fsn.getTimeLockBalance(  
  ↪ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff",  
  ↪ "0x9c48c796cb0bed51a14291bc8cc56ed7b5c")
```

## 6.25 makeSwap

```
fsn.makeSwap({from: fsn.coinbase, FromAssetID:  
  ↪ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff",  
  ToAssetID:  
  ↪ "0xffffffffffffffffffffffffffffffffffffffff00000000000000000000", MinToAmount: 1,  
  ↪ MinFromAmount: 2, SwapSize: 2, Targes: []}, "123456")  
...
```

makeSwap create a quantum swap CP see the top FromAssetID sell asset id MinFromAmount the min amount of the sell asset ToAssetID buy asset id MinToAmount the min amount of the buy asset SwapSize the max sell package size Targes the address list of the "who can buy" can be null password the owner password

## 6.25.1 Parameters

none

## 6.25.2 Returns

Object: With the following methods:

- Object: Description

## 6.25.3 Example

```
fsn.makeSwap({from: fsn.coinbase, FromAssetID:
↳ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff",
    ToAssetID:
↳ "0xffffffffffffffffffffffffffffffffffffffff00000000000000000000", MinToAmount: 1,
↳ MinFromAmount: 2, SwapSize: 2, Targes: []}, "123456")
```

## 6.26 recallSwap

```
fsn.recallSwap({from: fsn.coinbase, SwapID:
↳ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", "123456")
...

```

recallSwap destroy a quantum swap and get the asset back CP see the top SwapID the swap ID password the owner password

### 6.26.1 Parameters

none

### 6.26.2 Returns

Object: With the following methods:

- Object: Description

### 6.26.3 Example

```
fsn.recallSwap({from: fsn.coinbase, SwapID:
↳ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", "123456")
```

## 6.27 takeSwap

```
fnsn.takeSwap({from:fnsn.coinbase,SwapID:  
  ↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff",Size:"0x1"},  
  ↳"123456")  
...
```

takeSwap buy a quantum swap CP see the top SwapID the swap ID Size the package size password the owner password

### 6.27.1 Parameters

none

### 6.27.2 Returns

Object: With the following methods:

- Object: Description

### 6.27.3 Example

```
fnsn.takeSwap({from:fnsn.coinbase,SwapID:  
  ↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff",Size:"0x1"},  
  ↳"123456")
```

```
fnsn.getSnapshot( blocknumber )  
fnsn.getSnapshotHash(blockHas)
```

```
web3.fnsn.getHexDate = function( d ) {  
  return "0x" + (( new Date(d)).getTime() / 1000).toString(16)  
};  
  
web3.fnsn.hex2a = function( hexData ) {  
  hexData = hexData.replace( "0x", "" )  
  let hex = hexData.toString();//force conversion  
  let str = '';  
  for (let i = 0; (i < hex.length && hex.substr(i, 2) !== '00'); i += 2)  
    str += String.fromCharCode(parseInt(hex.substr(i, 2), 16));  
  return str;  
}
```

---

**Note:** This documentation is work in progress and web3-fusion-extend 1.0 stable is not yet released! You can find the current documentation for web3-fusion-extend 0.x.x at.

---



### FSNTX commands

- *Overview*
- *fsn*
- *Constants*

```
personal.unlockAccount( eth.coinbase, "123456" )
var tx = fsntx.buildBuyTicketTx( {from:eth.coinbase} )
tx.from = eth.coinbase
var tx2 = eth.signTransaction( tx )
fsntx.sendRawTransaction(tx2.tx)
```

## 7.1 sendRawTransaction

```
fsntx.sendRawTransaction()
...
```

sendRawTransaction

### 7.1.1 Parameters

none

### 7.1.2 Returns

Object: With the following methods:

- Object: Description

### 7.1.3 Example

```
fsntx.sendRawTransaction()
```

## 7.2 buildGenNotationTx

```
fsntx.buildGenNotationTx()  
...
```

buildGenNotationTx

### 7.2.1 Parameters

none

### 7.2.2 Returns

Object: With the following methods:

- Object: Description

### 7.2.3 Example

```
fsntx.buildGenNotationTx()
```

## 7.3 genNotation

```
fsntx.genNotation({from: fsn.coinbase}, "123456")  
...
```

genNotation gen a notation for a account CP see the top and the “from” ignore from who gen the notation password the account password

### 7.3.1 Parameters

none

### 7.3.2 Returns

Object: With the following methods:

- Object: Description

### 7.3.3 Example

```
fsntx.genNotation({from: fsn.coinbase}, "123456")
```

## 7.4 buildGenAssetTx

```
fsntx.buildGenAssetTx()  
...
```

buildGenAssetTx

### 7.4.1 Parameters

none

### 7.4.2 Returns

Object: With the following methods:

- Object: Description

### 7.4.3 Example

```
fsntx.buildGenAssetTx()
```

## 7.5 genAsset

```
fsntx.genAsset({from: fsn.coinbase, name: "FusionTest", symbol: "FST", decimals: 1, total:  
  ↳ "0x200"}, "123456")  
fsntx.genAsset({from: "0x91db50f5c36ae7616009d4e94462dca4d4c7e833", name: "JONESY",  
  ↳ symbol: "JSY", decimals: 1, total: "0x2000000000"}, "123123123")  
...
```

genAsset generate a asset CP see the top and the “from” ignore from who gen the asset and the owner of the asset name the name of asset symbol the symbol of asset decimals the asset decimal digit total the total number of the asset and the owner will be get same number asset CanChange whether asset can be incremented or decremented by the owner [optional] password the account password

### 7.5.1 Parameters

none

## 7.5.2 Returns

Object: With the following methods:

- Object: Description

## 7.5.3 Example

```
fsntx.genAsset({from:fsn.coinbase,name:"FusionTest",symbol:"FST",decimals:1,total:
↪"0x200"},"123456")
fsntx.genAsset({from:"0x91db50f5c36ae7616009d4e94462dca4d4c7e833",name:"JONESY",
↪symbol:"JSY",decimals:1,total:"0x2000000000"},"123123123")
```

## 7.6 buildSendAssetTx

```
fsntx.buildSendAssetTx()
...
```

buildSendAssetTx

### 7.6.1 Parameters

none

### 7.6.2 Returns

Object: With the following methods:

- Object: Description

### 7.6.3 Example

```
fsntx.buildSendAssetTx()
```

## 7.7 sendAsset

```
fsntx.sendAsset({from:fsn.coinbase,to:"0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165",
↪value:"0x1",asset:
↪"0x514a46f34e6eb0a98abb3595c4aec33ca8ddf69f135c8fed89e78d0808047965"},"123456")

fsntx.sendAsset({from:fsn.coinbase,to:"0x91db50F5c36aE7616009d4e94462DcA4D4c7e833",
↪value:"0x2",asset:
↪"0x88d18f81620e5684e880dddcf0b6c167a9154d4c499bc9fad47b98634110eeec"},"123456")
...
```

sendAsset send asset to other account CSP see the top

### 7.7.1 Parameters

none

### 7.7.2 Returns

Object: With the following methods:

- Object: Description

### 7.7.3 Example

```
fsntx.sendAsset({from: fsn.coinbase, to: "0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165",
↳ value: "0x1", asset:
↳ "0x514a46f34e6eb0a98abb3595c4aec33ca8ddf69f135c8fed89e78d0808047965"}, "123456")

fsntx.sendAsset({from: fsn.coinbase, to: "0x91db50F5c36aE7616009d4e94462DcA4D4c7e833",
↳ value: "0x2", asset:
↳ "0x88d18f81620e5684e880dddcf0b6c167a9154d4c499bc9fad47b98634110eeec"}, "123456")
```

## 7.8 buildAssetToTimeLockTx

```
fsntx.buildAssetToTimeLockTx()
...
```

buildAssetToTimeLockTx

### 7.8.1 Parameters

none

### 7.8.2 Returns

Object: With the following methods:

- Object: Description

### 7.8.3 Example

```
fsntx.buildAssetToTimeLockTx()
```

## 7.9 assetToTimeLock

```
fsntx.assetToTimeLock({asset :
↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.
↳coinbase, to: "0xa7455DF112c953F3c73c2C25559965e1A8a20024", start: "0x1", end: "0x2A300",
↳value: "0x1400000000000000"}, "123456")

fsntx.assetToTimeLock({asset :
↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.
↳coinbase, to: "0xa7455DF112c953F3c73c2C25559965e1A8a20024", start: "0x2a900", end:
↳"0x3A300", value: "0x1400000000000000"}, "123456")

fsntx.assetToTimeLock({asset :
↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.
↳coinbase, to: "0xa7455DF112c953F3c73c2C25559965e1A8a20024", start: getHexDate('2018-12-
↳01') , end: getHexDate('2019-01-01'), value: "0x1340000000000000"}, "123456")
...

```

assetToTimeLock send the asset to time lock CSP see the top startTime the start time of the time lock endTime the end time of the time lock password the account password

### 7.9.1 Parameters

none

### 7.9.2 Returns

Object: With the following methods:

- Object: Description

### 7.9.3 Example

```
fsntx.assetToTimeLock({asset :
↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.
↳coinbase, to: "0xa7455DF112c953F3c73c2C25559965e1A8a20024", start: "0x1", end: "0x2A300",
↳value: "0x1400000000000000"}, "123456")

fsntx.assetToTimeLock({asset :
↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.
↳coinbase, to: "0xa7455DF112c953F3c73c2C25559965e1A8a20024", start: "0x2a900", end:
↳"0x3A300", value: "0x1400000000000000"}, "123456")

fsntx.assetToTimeLock({asset :
↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.
↳coinbase, to: "0xa7455DF112c953F3c73c2C25559965e1A8a20024", start: getHexDate('2018-12-
↳01') , end: getHexDate('2019-01-01'), value: "0x1340000000000000"}, "123456")

```

## 7.10 buildTimeLockToTimeLockTx

```
fsntx.buildTimeLockToTimeLockTx()
...

```

buildTimeLockToTimeLockTx

### 7.10.1 Parameters

none

### 7.10.2 Returns

Object: With the following methods:

- Object: Description

### 7.10.3 Example

```
fsntx.buildTimeLockToTimeLockTx()
```

## 7.11 timeLockToTimeLock

```
fsntx.timeLockToTimeLock({asset:
  ↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.
  ↳coinbase, to: "0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165", start: "0x101", end: "0x200",
  ↳value: "0x100"}, "123456")
...

```

timeLockToTimeLock send the time lock CSP see the top startTime the start time of the time lock endTime the end time of the time lock password the account password

### 7.11.1 Parameters

none

### 7.11.2 Returns

Object: With the following methods:

- Object: Description

### 7.11.3 Example

```
fsntx.timeLockToTimeLock({asset:
  ↳"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.
  ↳coinbase, to: "0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165", start: "0x101", end: "0x200",
  ↳value: "0x100"}, "123456")

```

## 7.12 buildTimeLockToAssetTx

```
fsntx.buildTimeLockToAssetTx()
...

```

buildTimeLockToAssetTx

### 7.12.1 Parameters

none

### 7.12.2 Returns

Object: With the following methods:

- Object: Description

### 7.12.3 Example

```
fsntx.buildTimeLockToAssetTx()
```

## 7.13 timeLockToAsset

```
fsntx.timeLockToAsset ({asset :  
  ↳ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.  
  ↳ coinbase, to: fsn.coinbase, start: "0x0", end: "0x0", value: "0x100"}, "123456")  
...
```

timeLockToAsset send the time lock to asset CSP see the top startTime the start time of the time lock endTime the end time of the time lock password the account password

### 7.13.1 Parameters

none

### 7.13.2 Returns

Object: With the following methods:

- Object: Description

### 7.13.3 Example

```
fsntx.timeLockToAsset ({asset :  
  ↳ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", from: fsn.  
  ↳ coinbase, to: fsn.coinbase, start: "0x0", end: "0x0", value: "0x100"}, "123456")
```



## 7.14 buildBuyTicketTx

```
fsntx.buildBuyTicketTx()  
...
```

buildBuyTicketTx

### 7.14.1 Parameters

none

### 7.14.2 Returns

Object: With the following methods:

- Object: Description

### 7.14.3 Example

```
fsntx.buildBuyTicketTx()
```

## 7.15 buyTicket

```
fsntx.buyTicket({from: fsn.coinbase}, "123456")  
...
```

buyTicket buy the ticket CP see the top and the “from” ignore from who buy the ticket password the account password

### 7.15.1 Parameters

none

### 7.15.2 Returns

Object: With the following methods:

- Object: Description

### 7.15.3 Example

```
fsntx.buyTicket({from: fsn.coinbase}, "123456")
```

## 7.16 buildIncAssetTx

```
fsntx.buildIncAssetTx()  
...
```

buildIncAssetTx

### 7.16.1 Parameters

none

### 7.16.2 Returns

Object: With the following methods:

- Object: Description

### 7.16.3 Example

```
fsntx.buildIncAssetTx()
```

## 7.17 incAsset

```
fsntx.incAsset({from:fsn.coinbase,to:"0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165",  
↪value:"0x1",asset:  
↪"0x514a46f34e6eb0a98abb3595c4aec33ca8ddf69f135c8fed89e78d0808047965"}, "123456")  
...
```

incAsset increase account asset balance CSP see the top and the “from”,”to” ignore from the asset owner to the inc account password the account password

### 7.17.1 Parameters

none

### 7.17.2 Returns

Object: With the following methods:

- Object: Description

### 7.17.3 Example

```
fsntx.incAsset({from:fsn.coinbase,to:"0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165",  
↪value:"0x1",asset:  
↪"0x514a46f34e6eb0a98abb3595c4aec33ca8ddf69f135c8fed89e78d0808047965"}, "123456")
```

## 7.18 buildDecAssetTx

```
fsntx.buildDecAssetTx()
...
```

buildDecAssetTx

### 7.18.1 Parameters

none

### 7.18.2 Returns

Object: With the following methods:

- Object: Description

### 7.18.3 Example

```
fsntx.buildDecAssetTx()
```

## 7.19 decAsset

```
fsntx.decAsset ({from: fsn.coinbase, to: "0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165",
  ↳ value: "0x1", asset:
  ↳ "0x514a46f34e6eb0a98abb3595c4aec33ca8ddf69f135c8fed89e78d0808047965"}, "123456")
...
```

decAsset decrease account asset balance CSP see the top and the “from”,”to” ignore from the asset owner to the dec account password the account password

### 7.19.1 Parameters

none

### 7.19.2 Returns

Object: With the following methods:

- Object: Description

### 7.19.3 Example

```
fsntx.decAsset ({from: fsn.coinbase, to: "0x2b1a3eca81ba03a9a4c95f4a04679c90838d7165",
  ↳ value: "0x1", asset:
  ↳ "0x514a46f34e6eb0a98abb3595c4aec33ca8ddf69f135c8fed89e78d0808047965"}, "123456")
```

## 7.20 buildMakeSwapTx

```
fsntx.buildMakeSwapTx()  
...
```

buildMakeSwapTx

### 7.20.1 Parameters

none

### 7.20.2 Returns

Object: With the following methods:

- Object: Description

### 7.20.3 Example

```
fsntx.buildMakeSwapTx()
```

## 7.21 makeSwap

```
fsntx.makeSwap({from: fsn.coinbase, FromAssetID:  
↪ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff",  
  ToAssetID:  
↪ "0xffffffffffffffffffffffffffffffffffffffff00000000000000000000", MinToAmount: 1,  
↪ MinFromAmount: 2, SwapSize: 2, Targes: []}, "123456")  
...
```

makeSwap create a quantum swap CP see the top FromAssetID sell asset id MinFromAmount the min amount of the sell asset ToAssetID buy asset id MinToAmount the min amount of the buy asset SwapSize the max sell package size Targes the address list of the “who can buy” can be null password the owner password

### 7.21.1 Parameters

none

### 7.21.2 Returns

Object: With the following methods:

- Object: Description

### 7.21.3 Example

```
fsntx.makeSwap({from:fsn.coinbase,FromAssetID:
↪"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff",
    ToAssetID:
↪"0xffffffffffffffffffffffffffffffffffffffff00000000000000000000",MinToAmount:1,
↪MinFromAmount:2,SwapSize:2,Targes:[]},"123456")
```

## 7.22 buildRecallSwapTx

```
fsntx.buildRecallSwapTx()
...
```

buildRecallSwapTx

### 7.22.1 Parameters

none

### 7.22.2 Returns

Object: With the following methods:

- Object: Description

### 7.22.3 Example

```
fsntx.buildRecallSwapTx()
```

## 7.23 recallSwap

```
fsntx.recallSwap({from:fsn.coinbase,SwapID:
↪"0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff"},"123456")
...
```

recallSwap destroy a quantum swap and get the asset back CP see the top SwapID the swap ID password the owner password

### 7.23.1 Parameters

none

### 7.23.2 Returns

Object: With the following methods:

- Object: Description

### 7.23.3 Example

```
fsntx.recallSwap({from: fsn.coinbase, SwapID:  
  ↪ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", "123456")
```

## 7.24 buildTakeSwapTx

```
fsntx.buildTakeSwapTx()  
...
```

buildTakeSwapTx

### 7.24.1 Parameters

none

### 7.24.2 Returns

Object: With the following methods:

- Object: Description

### 7.24.3 Example

```
fsntx.buildTakeSwapTx()
```

## 7.25 takeSwap

```
fsntx.takeSwap({from: fsn.coinbase, SwapID:  
  ↪ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", Size: "0x1"},  
  ↪ "123456")  
...
```

takeSwap buy a quantum swap CP see the top SwapID the swap ID Size the package size password the owner password

### 7.25.1 Parameters

none

### 7.25.2 Returns

Object: With the following methods:

- Object: Description

### 7.25.3 Example

```
fsntx.takeSwap({from: fsn.coinbase, SwapID:  
  ↪ "0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff", Size: "0x1"},  
  ↪ "123456")
```





## A

autoPurchaseTickets, 11

## B

blockexplorerapi, 13

## C

constants, 17

Contributing, 3

## F

fsn, 21

fsntx, 37

## G

Getting Started, 7