



BaseUrl – <http://localhost:7292/>

<https://localhost:7777/>

Port – Port is in config file. Can be changed but defaults to 7292 or if test URL enabled it is 7777.

Passwords – If you see `{somePassword?}` in the URL that is only required when the user has setup in the config to always require a password for API calls. If you are not using that or you are using the wallet timed locks then you will not need to pass the password and would use the main URL above it.

## Table of Contents

Beacon API - Methods .....	5
CreateBeacon .....	5
DecodeBeaconLocator .....	6
GetBeaconInfo.....	6
SetBeaconState .....	7
Smart Contract API .....	8
SCV1 .....	9
SCPassTest.....	10
SCPassDesTest.....	10
GetAllSmartContracts.....	10
GetMintedSmartContracts .....	11
GetSingleSmartContract.....	11
CreateSmartContract .....	12
MintSmartContract .....	12
ChangeNFTPublicState .....	12
GetSmartContractData.....	13
TransferNFT.....	13
Burn .....	13
Evolve .....	14
Devolve.....	14
EvolveSpecific .....	15
Transaction API.....	16
GetTimeStamp.....	16
GetAddressNonce.....	17
GetRawTXFee .....	17
GetTxHash .....	18
ValidateSignature .....	18
VerifyRawTransaction .....	18
SendRawTransaction .....	19
TestRawTransaction .....	19

---

CreateDnr .....	19
SendTransaction .....	20
Main API (v1) .....	21
UnlockWallet .....	25
LockWallet .....	25
CheckStatus .....	25
GetHDWallet .....	26
GetRestoreHDWallet .....	26
GetNewAddress .....	26
GetWalletInfo .....	27
GetAllAddresses .....	27
GetValidatorAddresses .....	27
TurnOnValidator .....	28
TurnOffValidator .....	28
GetValidatorInfo .....	28
GetAddressInfo .....	29
ImportPrivateKey .....	29
SendBlock .....	29
GetLastBlock .....	30
GetAllTransactions .....	30
StartValidating .....	31
ChangeValidatorName .....	31
CreateSignature .....	31
ValidateSignature .....	32
GetMempool .....	32
GetMemBlockCluster .....	33
GetTaskAnswerList .....	33
GetMasternodesSent .....	33
GetMasternodes .....	34
GetValidatorPoolInfo .....	34
GetPeerInfo .....	34

---

---

GetDebugInfo .....	35
GetCLIVersion .....	35
ReadRBXLog .....	36
ClearRBXLog .....	36
ReadValLog .....	36
ClearValLog .....	36
SendExit .....	37
SendExitComplete .....	37
Models .....	38
Account .....	38
Adjudicators .....	38
Adnr .....	39
BeaconInfo .....	39
Block .....	40
HDWallet .....	40
Peers .....	41
TaskAnswer .....	41
TaskQuestion .....	41
Transaction .....	42
Validators .....	43

## Beacon API - Methods

### BCV1



**GET** /bcapi/BCV1/CreateBeacon/{name}



**GET** /bcapi/BCV1/{somePassword}/CreateBeacon/{name}



**GET** /bcapi/BCV1/DecodeBeaconLocator/{locator}



**GET** /bcapi/BCV1/{somePassword}/DecodeBeaconLocator/{locator}



**GET** /bcapi/BCV1/GetBeaconInfo



**GET** /bcapi/BCV1/{somePassword}/GetBeaconInfo



**GET** /bcapi/BCV1/SetBeaconState



**GET** /bcapi/BCV1/{somePassword}/SetBeaconState



### CreateBeacon

Creates a beacon locator on the local host.

**GET**`/bcapi/BCV1/CreateBeacon/{name}`**Parameters**

Name	Description
<b>name</b> * required string (path)	<input type="text" value="name"/>

**DecodeBeaconLocator**

Decodes a given beacon locator

**GET**`/bcapi/BCV1/DecodeBeaconLocator/{locator}`**Parameters**

Name	Description
<b>locator</b> * required string (path)	<input type="text" value="locator"/>

**GetBeaconInfo**

Returns the local beacon info for the user

**GET****/bcapi/BCV1/GetBeaconInfo****Parameters**

No parameters

**Responses****SetBeaconState**

Toggles the beacons state to on and off

**GET****/bcapi/BCV1/SetBeaconState****Parameters**

No parameters

## Smart Contract API

<b>GET</b>	/scapi/SCV1	▼
<b>GET</b>	/scapi/SCV1/{somePassword}	▼
<b>GET</b>	/scapi/SCV1/{id}	▼
<b>GET</b>	/scapi/SCV1/{somePassword}/{id}	▼
<b>POST</b>	/scapi/SCV1/SCPAssTest	▼
<b>POST</b>	/scapi/SCV1/{somePassword}/SCPAssTest	▼
<b>POST</b>	/scapi/SCV1/SCPAssDesTest	▼
<b>POST</b>	/scapi/SCV1/{somePassword}/SCPAssDesTest	▼
<b>GET</b>	/scapi/SCV1/GetAllSmartContracts	▼
<b>GET</b>	/scapi/SCV1/{somePassword}/GetAllSmartContracts	▼
<b>GET</b>	/scapi/SCV1/GetMintedSmartContracts	▼
<b>GET</b>	/scapi/SCV1/{somePassword}/GetMintedSmartContracts	▼
<b>GET</b>	/scapi/SCV1/GetSingleSmartContract/{id}	▼
<b>GET</b>	/scapi/SCV1/{somePassword}/GetSingleSmartContract/{id}	▼
<b>POST</b>	/scapi/SCV1/CreateSmartContract	▼
<b>POST</b>	/scapi/SCV1/{somePassword}/CreateSmartContract	▼
<b>GET</b>	/scapi/SCV1/MintSmartContract/{id}	▼

GET	/scapi/SCV1/{somePassword}/MintSmartContract/{id}	✓
GET	/scapi/SCV1/ChangeNFTPublicState/{id}	✓
GET	/scapi/SCV1/{somePassword}/ChangeNFTPublicState/{id}	✓
GET	/scapi/SCV1/GetSmartContractData/{id}	✓
GET	/scapi/SCV1/{somePassword}/GetSmartContractData/{id}	✓
GET	/scapi/SCV1/TestDynamicNFT/{id}	✓
GET	/scapi/SCV1/{somePassword}/TestDynamicNFT/{id}	✓
GET	/scapi/SCV1/TestRemove/{id}/{toAddress}	✓
GET	/scapi/SCV1/{somePassword}/TestRemove/{id}/{toAddress}	✓
GET	/scapi/SCV1/TransferNFT/{id}/{toAddress}	✓
GET	/scapi/SCV1/{somePassword}/TransferNFT/{id}/{toAddress}	✓
GET	/scapi/SCV1/Burn/{id}	✓
GET	/scapi/SCV1/{somePassword}/Burn/{id}	✓
GET	/scapi/SCV1/Evolve/{id}/{toAddress}	✓
GET	/scapi/SCV1/{somePassword}/Evolve/{id}/{toAddress}	✓
GET	/scapi/SCV1/Devolve/{id}/{toAddress}	✓
GET	/scapi/SCV1/{somePassword}/Devolve/{id}/{toAddress}	✓
GET	/scapi/SCV1/EvolveSpecific/{id}/{toAddress}/{evolveState}	✓

## SCV1

Returns the API is online.

GET /scapi/SCV1

### Parameters

No parameters

## SCPassTest

This returns exactly what the user passed to API. This is a test to verify things are working. It is expecting a type of application/json.

POST /scapi/SCV1/SCPassTest

### Parameters

Try it out

No parameters

Request body

application/json

## SCPassDesTest

This returns exactly what the user passed to API, but it will first attempt to deserialize it against the SmartContractMain model. This is a test to verify things are working. It is expecting a type of application/json.

POST /scapi/SCV1/SCPassDesTest

### Parameters

Try it out

No parameters

Request body

application/json

## GetAllSmartContracts

Returns all smart contracts that are locally owned by client.

**GET** /scapi/SCV1/GetAllSmartContracts

#### Parameters

No parameters

### GetMintedSmartContracts

This returns all smart contracts where the minter is present in the wallet accounts.

**GET** /scapi/SCV1/GetMintedSmartContracts

#### Parameters

No parameters

### GetSingleSmartContract

Returns a single smart contract with the provided smart contract UID.

**GET** /scapi/SCV1/GetSingleSmartContract/{id}

#### Parameters

Name	Description
------	-------------

<b>id</b> * required	
----------------------	--

string	
--------	--

(path)	
--------	--

id
----

## CreateSmartContract

Creates a smart contract when passed a json model of SmartContractMain. This will create the contract locally, but not mint it. That happens next.

The screenshot shows a POST request to /scapi/SCV1/CreateSmartContract. The 'Parameters' section indicates 'No parameters'. The 'Request body' section is set to application/json. A 'Try it out' button is visible in the top right corner.

## MintSmartContract

Takes the smart contract UID and mints it to the network.

The screenshot shows a GET request to /scapi/SCV1/MintSmartContract/{id}. The 'Parameters' section includes a required parameter 'id' of type string (path). The 'Name' column lists 'id' and the 'Description' column lists 'id'.

## ChangeNFTPublicState

Changes the public state of an NFT.

The screenshot shows a GET request to /scapi/SCV1/ChangeNFTPublicState/{id}. The 'Parameters' section includes a required parameter 'id' of type string (path). The 'Name' column lists 'id' and the 'Description' column lists 'id'.

## GetSmartContractData

Gets any smart contract minted on the network with smart contract UID.

GET

/scapi/SCV1/GetSmartContractData/{id}

### Parameters

Name	Description
------	-------------

**id** \* required  
string  
(path)

id

## TransferNFT

Transfer an NFT with a smart contract UID and a To Address

GET

/scapi/SCV1/TransferNFT/{id}/{toAddress}

### Parameters

Name	Description
------	-------------

**id** \* required  
string  
(path)

id

**toAddress** \* required  
string  
(path)

toAddress

## Burn

Burns an NFT with provided smart contract UID. You must own this NFT.

**GET****/scapi/SCV1/Burn/{id}**

### Parameters

Name	Description
------	-------------

<b>id</b> * required	
----------------------	--

string	
--------	--

(path)	
--------	--

id	
----	--

## Evolve

Evolves an NFT to the next level. Can never go beyond the highest level.

### Parameters

Name	Description
------	-------------

<b>id</b> * required	
----------------------	--

string	
--------	--

(path)	
--------	--

id	
----	--

<b>toAddress</b> * required	
-----------------------------	--

string	
--------	--

(path)	
--------	--

toAddress	
-----------	--

## Devolve

Devolves an NFT down to the next level. Can never go beyond the lowest level.

**GET** /scapi/SCV1/Devolve/{id}/{toAddress}

**Parameters**

Name	Description
<b>id</b> * required string (path)	id
<b>toAddress</b> * required string (path)	toAddress

## EvolveSpecific

Evolves/Devolves to a specific level. Cannot go above or below the min/max states.

**GET** /scapi/SCV1/EvolveSpecific/{id}/{toAddress}/{evolveState}

**Parameters**

Name	Description
<b>id</b> * required string (path)	id
<b>toAddress</b> * required string (path)	toAddress
<b>evolveState</b> * required integer(\$int32) (path)	evolveState

## Transaction API

<code>GET</code>	/txapi/TXV1/GetTimestamp	▼
<code>GET</code>	/txapi/TXV1/{somePassword}/GetTimestamp	▼
<code>GET</code>	/txapi/TXV1/GetAddressNonce/{address}	▼
<code>GET</code>	/txapi/TXV1/{somePassword}/GetAddressNonce/{address}	▼
<code>POST</code>	/txapi/TXV1/GetRawTxFee	▼
<code>POST</code>	/txapi/TXV1/{somePassword}/GetRawTxFee	▼
<code>POST</code>	/txapi/TXV1/GetTxHash	▼
<code>POST</code>	/txapi/TXV1/{somePassword}/GetTxHash	▼
<code>GET</code>	/txapi/TXV1/ValidateSignature/{message}/{address} /{sigScript}	▼
<code>GET</code>	/txapi/TXV1/{somePassword}/ValidateSignature/{message} /{address}/{sigScript}	▼
<code>POST</code>	/txapi/TXV1/VerifyRawTransaction	▼
<code>POST</code>	/txapi/TXV1/{somePassword}/VerifyRawTransaction	▼
<code>POST</code>	/txapi/TXV1/SendRawTransaction	▼
<code>POST</code>	/txapi/TXV1/{somePassword}/SendRawTransaction	▼
<code>POST</code>	/txapi/TXV1/TestRawTransaction	▼
<code>POST</code>	/txapi/TXV1/{somePassword}/TestRawTransaction	▼
<code>GET</code>	/txapi/TXV1/CreateDnr/{address}/{name}	▼
<code>GET</code>	/txapi/TXV1/{somePassword}/CreateDnr/{address}/{name}	▼
<code>GET</code>	/txapi/TXV1/SendTransaction/{faddr}/{taddr}/{amt}	▼
<code>GET</code>	/txapi/TXV1/{somePassword}/SendTransaction/{faddr}/{taddr} /{amt}	▼

### GetTimeStamp

Returns the current timestamp

**GET** /txapi/TXV1/GetTimestamp

#### Parameters

No parameters

### GetAddressNonce

Returns the next nonce for an account owned address.

**GET** /txapi/TXV1/GetAddressNonce/{address}

#### Parameters

Name	Description
address * required string (path)	address

### GetRawTxFee

Returns the RAW tx fee for a transaction. This expects a json body of the model Transaction

**POST** /txapi/TXV1/GetRawTxFee

#### Parameters

Try it out ^

No parameters

Request body

application/json

## GetTxHash

Gets the Hash from an expected json body of a Transaction

The screenshot shows a POST request to /txapi/TXV1/GetTxHash. The 'Parameters' section indicates 'No parameters'. The 'Request body' section is set to 'application/json'. A 'Try it out' button is visible in the top right.

## ValidateSignature

Validates a given signature to determine if the sign is valid.

The screenshot shows a GET request to /txapi/TXV1/ValidateSignature/{message}/{address}/{sigScript}. The 'Parameters' section includes three fields: 'message' (string, path), 'address' (string, path), and 'sigScript' (string, path). Each field has a corresponding input field for entering values.

## VerifyRawTransaction

This expects a fully complete json body of a transaction. If complete it will validate that it is a valid transaction for publication to mempool.

POST /txapi/TXV1/VerifyRawTransaction

Parameters

No parameters

Request body

application/json

## SendRawTransaction

This will send a raw TX to the mempool for publication

POST /txapi/TXV1/SendRawTransaction

Parameters

No parameters

Request body

application/json

## TestRawTransaction

This will take your json body of a Transaction and attempt to deserialize it and then re-serialize it. This method is good to test your body and ensure all needed things are there.

POST /txapi/TXV1/TestRawTransaction

Parameters

No parameters

Request body

application/json

## CreateDnr

This creates a domain name address associated request.

**GET** /txapi/TXV1/CreateDnr/{address}/{name}

**Parameters**

Name	Description
<b>address</b> * required string (path)	address
<b>name</b> * required string (path)	name

## SendTransaction

This sends a base RBX transaction for just coins.

**GET** /txapi/TXV1/SendTransaction/{faddr}/{taddr}/{amt}

**Parameters**

Name	Description
<b>faddr</b> * required string (path)	faddr
<b>taddr</b> * required string (path)	taddr
<b>amt</b> * required string (path)	amt

## Main API (v1)

<b>GET</b>	/api/V1	▼
<b>GET</b>	/api/V1/{somePassword}	▼
<b>GET</b>	/api/V1/{id}	▼
<b>GET</b>	/api/V1/{somePassword}/{id}	▼
<b>GET</b>	/api/V1/UnlockWallet/{password}	▼
<b>GET</b>	/api/V1/{somePassword}/UnlockWallet/{password}	▼
<b>GET</b>	/api/V1/LockWallet/{password}	▼
<b>GET</b>	/api/V1/{somePassword}/LockWallet/{password}	▼
<b>GET</b>	/api/V1/CheckStatus	▼
<b>GET</b>	/api/V1/{somePassword}/CheckStatus	▼
<b>GET</b>	/api/V1/GetHDWallet/{strength}	▼
<b>GET</b>	/api/V1/{somePassword}/GetHDWallet/{strength}	▼
<b>GET</b>	/api/V1/GetRestoreHDWallet/{mnemonic}	▼
<b>GET</b>	/api/V1/{somePassword}/GetRestoreHDWallet/{mnemonic}	▼
<b>GET</b>	/api/V1/GetNewAddress	▼
<b>GET</b>	/api/V1/{somePassword}/GetNewAddress	▼
<b>GET</b>	/api/V1/GetWalletInfo	▼
<b>GET</b>	/api/V1/{somePassword}/GetWalletInfo	▼
<b>GET</b>	/api/V1/GetAllAddresses	▼
<b>GET</b>	/api/V1/{somePassword}/GetAllAddresses	▼

<b>GET</b>	/api/V1/GetValidatorAddresses	▼
<b>GET</b>	/api/V1/{somePassword}/GetValidatorAddresses	▼
<b>GET</b>	/api/V1/TurnOnValidator/{id}	▼
<b>GET</b>	/api/V1/{somePassword}/TurnOnValidator/{id}	▼
<b>GET</b>	/api/V1/TurnOffValidator/{id}	▼
<b>GET</b>	/api/V1/{somePassword}/TurnOffValidator/{id}	▼
<b>GET</b>	/api/V1/GetValidatorInfo/{id}	▼
<b>GET</b>	/api/V1/{somePassword}/GetValidatorInfo/{id}	▼
<b>GET</b>	/api/V1/GetAddressInfo/{id}	▼
<b>GET</b>	/api/V1/{somePassword}/GetAddressInfo/{id}	▼
<b>GET</b>	/api/V1/ImportPrivateKey/{id}	▼
<b>GET</b>	/api/V1/{somePassword}/ImportPrivateKey/{id}	▼
<b>GET</b>	/api/V1/SendBlock/{id}	▼
<b>GET</b>	/api/V1/{somePassword}/SendBlock/{id}	▼
<b>GET</b>	/api/V1/GetLastBlock	▼
<b>GET</b>	/api/V1/{somePassword}/GetLastBlock	▼
<b>GET</b>	/api/V1/GetRollbackBlocks/{id}	▼
<b>GET</b>	/api/V1/{somePassword}/GetRollbackBlocks/{id}	▼
<b>GET</b>	/api/V1/GetAllTransactions	▼
<b>GET</b>	/api/V1/{somePassword}/GetAllTransactions	▼
<b>GET</b>	/api/V1/SendTransaction/{faddr}/{taddr}/{amt}	▼

<b>GET</b>	/api/V1/{somePassword}/SendTransaction/{faddr}/{taddr}/{amt}	▼
<b>GET</b>	/api/V1/StartValidating/{addr}/{uname}	▼
<b>GET</b>	/api/V1/{somePassword}/StartValidating/{addr}/{uname}	▼
<b>GET</b>	/api/V1/ChangeValidatorName/{uname}	▼
<b>GET</b>	/api/V1/{somePassword}/ChangeValidatorName/{uname}	▼
<b>GET</b>	/api/V1/CreateSignature/{message}/{address}	▼
<b>GET</b>	/api/V1/{somePassword}/CreateSignature/{message}/{address}	▼
<b>GET</b>	/api/V1/ValidateSignature/{message}/{address}/{sigScript}	▼
<b>GET</b>	/api/V1/{somePassword}/ValidateSignature/{message}/{address}/{sigScript}	▼
<b>GET</b>	/api/V1/GetMempool	▼
<b>GET</b>	/api/V1/{somePassword}/GetMempool	▼
<b>GET</b>	/api/V1/GetMemBlockCluster	▼
<b>GET</b>	/api/V1/{somePassword}/GetMemBlockCluster	▼
<b>GET</b>	/api/V1/GetTaskAnswersList	▼
<b>GET</b>	/api/V1/{somePassword}/GetTaskAnswersList	▼
<b>GET</b>	/api/V1/GetMasternodesSent	▼
<b>GET</b>	/api/V1/{somePassword}/GetMasternodesSent	▼
<b>GET</b>	/api/V1/GetMasternodes	▼
<b>GET</b>	/api/V1/{somePassword}/GetMasternodes	▼
<b>GET</b>	/api/V1/GetValidatorPoolInfo	▼
<b>GET</b>	/api/V1/{somePassword}/GetValidatorPoolInfo	▼

<b>GET</b>	/api/V1/{somePassword}/GetValidatorPoolInfo	▼
<b>GET</b>	/api/V1/GetPeerInfo	▼
<b>GET</b>	/api/V1/{somePassword}/GetPeerInfo	▼
<b>GET</b>	/api/V1/GetDebugInfo	▼
<b>GET</b>	/api/V1/{somePassword}/GetDebugInfo	▼
<b>GET</b>	/api/V1/GetCLIVersion	▼
<b>GET</b>	/api/V1/{somePassword}/GetCLIVersion	▼
<b>GET</b>	/api/V1/ReadRBXLog	▼
<b>GET</b>	/api/V1/{somePassword}/ReadRBXLog	▼
<b>GET</b>	/api/V1/ClearRBXLog	▼
<b>GET</b>	/api/V1/{somePassword}/ClearRBXLog	▼
<b>GET</b>	/api/V1/ReadValLog	▼
<b>GET</b>	/api/V1/{somePassword}/ReadValLog	▼
<b>GET</b>	/api/V1/ClearValLog	▼
<b>GET</b>	/api/V1/{somePassword}/ClearValLog	▼
<b>GET</b>	/api/V1/SendExit	▼
<b>GET</b>	/api/V1/{somePassword}/SendExit	▼
<b>GET</b>	/api/V1/SendExitComplete	▼
<b>GET</b>	/api/V1/{somePassword}/SendExitComplete	▼

## UnlockWallet

This will unlock the wallet for the configured amount of time. If not set default time is 15 minutes.

GET /api/V1/UnlockWallet/{password}

**Parameters**

Name	Description
<b>password</b> * required string (path)	password

## LockWallet

This will lock the wallet back even if the unlock time has not passed.

GET /api/V1/LockWallet/{password}

**Parameters**

Name	Description
<b>password</b> * required string (path)	password

## CheckStatus

Checks to see if API is online. Will return ‘Online’ if it is.

GET /api/V1/CheckStatus

**Parameters**

No parameters

## GetHDWallet

This will return a Mnemonic. It requires a strength of 12 or 24.

**GET** /api/V1/GetHDWallet/{strength}

**Parameters**

Name	Description
<b>strength</b> * required integer(\$int32) (path)	strength

## GetRestoreHDWallet

This will restore a wallet with a supplied mnemonic.

**GET** /api/V1/GetRestoreHDWallet/{mnemonic}

**Parameters**

Name	Description
<b>mnemonic</b> * required string (path)	mnemonic

## GetNewAddress

This will return a new address for you to use immediately.

**GET** /api/V1/GetNewAddress

**Parameters**

No parameters
---------------

## GetWalletInfo

This is a method used to pass information to GUI and other applications.

GET	/api/V1/GetWalletInfo
<b>Parameters</b>	
No parameters	

[{"BlockHeight":"80753","PeerCount":"3","BlocksDownloading":"True","IsResyncing":"False","IsChainSynced":"False","ChainCorrupted":"False"}]

## GetAllAddresses

This will return all locally owned addresses

GET	/api/V1/GetAllAddresses
<b>Parameters</b>	
No parameters	

## GetValidatorAddresses

This is for GUI applications and it will return all eligible validator accounts.

GET	/api/V1/GetValidatorAddresses
<b>Parameters</b>	
No parameters	

## TurnOnValidator

This will turn a validator back to online and allow it to rejoin the pool. The ID is the validator address.

GET	/api/V1/TurnOnValidator/{id}
-----	------------------------------

**Parameters**

Name	Description
<b>id</b> * required string (path)	id

## TurnOffValidator

This will turn off a validator and allow them to exit a pool. The ID is the validator address

GET	/api/V1/TurnOffValidator/{id}
-----	-------------------------------

**Parameters**

Name	Description
<b>id</b> * required string (path)	id

## GetValidatorInfo

This will return the current validator's name if present.

GET	/api/V1/GetValidatorInfo/{id}
-----	-------------------------------

**Parameters**

Name	Description
<b>id</b> * required string (path)	id

## GetAddressInfo

This will return the model information for an address account. The ID is the address

GET	/api/V1/GetAddressInfo/{id}
-----	-----------------------------

**Parameters**

Name	Description
<b>id</b> * required string (path)	id

## ImportPrivateKey

This allows you to add an account to your local wallet with a supplied private key.

GET	/api/V1/ImportPrivateKey/{id}
-----	-------------------------------

**Parameters**

Name	Description
<b>id</b> * required string (path)	id

## SendBlock

This will allow you to get information of a block with its height.

GET /api/V1/SendBlock/{id}

#### Parameters

Name	Description
------	-------------

<b>id</b> * required string (path)	id
--	----

## GetLastBlock

This will return the most recent block you have synced.

GET /api/V1/GetLastBlock

#### Parameters

No parameters

## GetAllTransactions

This will return all transactions associated with this wallet.

GET /api/V1/GetAllTransactions

#### Parameters

No parameters

## StartValidating

This will allow you to begin validating if you have never registered an address from the wallet.

**GET** /api/V1/StartValidating/{addr}/{uname}

Parameters	
Name	Description
<b>addr</b> * required string (path)	addr
<b>uname</b> * required string (path)	uname

## ChangeValidatorName

This will allow you to change your validator name.

**GET** /api/V1/ChangeValidatorName/{uname}

Parameters	
Name	Description
<b>uname</b> * required string (path)	uname

## CreateSignature

This lets you choose an address and sign whatever message you like.

**GET** /api/V1/CreateSignature/{message}/{address}

**Parameters**

Name	Description
<b>message</b> * required string (path)	message
<b>address</b> * required string (path)	address

## ValidateSignature

This lets you take any message, signature, and address and validate the sign.

**GET** /api/V1/ValidateSignature/{message}/{address}/{sigScript}

**Parameters**

Name	Description
<b>message</b> * required string (path)	message
<b>address</b> * required string (path)	address
<b>sigScript</b> * required string (path)	sigScript

## GetMempool

This will return the current mempool.

GET	/api/V1/GetMempool
-----	--------------------

#### Parameters

No parameters

## GetMemBlockCluster

This returns a cluster of blocks that are stored in memory.

GET	/api/V1/GetMemBlockCluster
-----	----------------------------

#### Parameters

No parameters

## GetTaskAnswerList

This will return a list of task answers supplied to lead adj.

GET	/api/V1/GetTaskAnswersList
-----	----------------------------

#### Parameters

No parameters

## GetMasternodesSent

This will return a list of masternodes that are fresh within 15 minutes.

**GET** /api/V1/GetMasternodesSent

#### Parameters

No parameters

## GetMasternodes

This will return a list of masternodes.

**GET** /api/V1/GetMasternodes

#### Parameters

No parameters

## GetValidatorPoolInfo

This will let you know if you are connected to an adjudicator and when you connected.

**GET** /api/V1/GetValidatorPoolInfo

#### Parameters

No parameters

## GetPeerInfo

This will return the information for all peers you are connected too

GET	/api/V1/GetPeerInfo
-----	---------------------

#### Parameters

No parameters

## GetDebugInfo

This will return a print out of ready at hand information

GET	/api/V1/GetDebugInfo
-----	----------------------

#### Parameters

No parameters

## GetCLIVersion

This will return the current version of the CLI.

GET	/api/V1/GetCLIVersion
-----	-----------------------

#### Parameters

No parameters

Major Version -> Minor Version -> Days since first wallet ->(optional pre if in pre-release)

1.22.189-pre

## ReadRBXLog

This will print out the data contained in the RBX log.

**GET** /api/V1/ReadRBXLog

### Parameters

No parameters

## ClearRBXLog

This will clear the RBX log.

**GET** /api/V1/ClearRBXLog

### Parameters

No parameters

## ReadValLog

This will read and print out the validator log file.

**GET** /api/V1/ReadValLog

### Parameters

No parameters

## ClearValLog

Clears the validator log file

GET	/api/V1/ClearValLog
-----	---------------------

#### Parameters

No parameters

## SendExit

This will send a command to CLI to stop all timers and then within 1 second of that exit the environment

GET	/api/V1/SendExit
-----	------------------

#### Parameters

No parameters

## SendExitComplete

After send exit call this method. If nothing is returned then wallet exited. If you receive 'SA' then wallet has not closed.

GET	/api/V1/SendExitComplete
-----	--------------------------

#### Parameters

No parameters

## Models

### Account

```
0 references
public long Id { get; set; }
19 references
public string PrivateKey { set; get; }
20 references
public string PublicKey { set; get; }
72 references
public string Address { get; set; }
20 references
public decimal Balance { get; set; }
21 references
public bool IsValidating { get; set; }
0 references
public bool IsEncrypted { get; set; }
```

### Adjudicators

```
0 references
public long Id { get; set; }
3 references
public string Address { get; set; }
1 reference
public string UniqueName { get; set; }
1 reference
public string Signature { get; set; }
1 reference
public bool IsActive { get; set; }
8 references
public bool IsLeadAdjuidcator { get; set; }
3 references
public string NodeIP { get; set; }
1 reference
public string WalletVersion { get; set; }
1 reference
public DateTime LastChecked { get; set; }
```

**Adnr**

```
public int Id { get; set; }

public string Address { get; set; }

public string Signature { get; set; }

public string Name { get; set; }

public string Hash { get; set; }

public string TxHash { get; set; }

public long Timestamp { get; set; }
```

**BeaconInfo**

```
public int Id { get; set; }

public string Name { get; set; }

public string BeaconLocator { get; set; }

public bool IsBeaconActive { get; set; }

public string BeaconUID { get; set; }
```

## Block

```

82 references
public long Height { get; set; }
17 references
public string ChainRefId { get; set; }
21 references
public long Timestamp { get; set; }
24 references
public string Hash { get; set; }
5 references
public string PrevHash { get; set; }
11 references
public string MerkleRoot { get; set; }
11 references
public string StateRoot { get; set; }
19 references
public string Validator { get; set; }
5 references
public string ValidatorSignature { get; set; }
7 references
public string ValidatorAnswer { get; set; }
3 references
public decimal TotalReward { get; set; }
7 references
public int TotalValidators { get; set; }
3 references
public decimal TotalAmount { get; set; }
6 references
public int Version { get; set; }
4 references
public int NumOfTx { get; set; }
3 references
public long Size { get; set; }
3 references
public int BCraftTime { get; set; }

```

## HDWallet

```

0 references
public int Id { get; set; }
3 references
public string WalletSeed { get; set; }
4 references
public int Nonce { get; set; }
3 references
public string Path { get; set; }

```

### Peers

```
0 references
public long Id { get; set; }
14 references
public string PeerIP { get; set; }
4 references
public bool IsIncoming { get; set; }
6 references
public bool IsOutgoing { get; set; }
7 references
public int FailCount { get; set; }
0 references
```

### TaskAnswer

```
7 references
public string Address { get; set; }
6 references
public string Answer { get; set; }
8 references
public Block Block { get; set; }
5 references
public DateTime SubmitTime { get; set; }
```

### TaskQuestion

```
public string TaskAnswer { get; set; }

public string TaskType { get; set; }

public long BlockHeight { get; set; }
```

## Transaction

```
[StringLength(128)]  
99+ references  
public string Hash { get; set; }  
[StringLength(36)]  
47 references  
public string ToAddress { get; set; }  
[StringLength(36)]  
77 references  
public string FromAddress { get; set; }  
69 references  
public decimal Amount { get; set; }  
23 references  
public long Nonce { get; set; }  
68 references  
public decimal Fee { get; set; }  
36 references  
public long Timestamp { get; set; }  
43 references  
public string? Data { get; set; }  
[StringLength(512)]  
13 references  
public string Signature { get; set; }  
2 references  
public long Height { get; set; }  
43 references  
public TransactionType TransactionType { get; set; }
```

## Validators

```
3 references
public long Id { get; set; }

26 references
public string Address { get; set; }

9 references
public string UniqueName { get; set; }

9 references
public long Position { get; set; }

3 references
public decimal Amount { get; set; } //Must be 1000 or more.

3 references
public long EligibleBlockStart { get; set; }

3 references
public string Signature { get; set; }

7 references
public bool IsActive { get; set; }

6 references
public int FailCount { get; set; }

7 references
public string NodeIP { get; set; }

5 references
public string NodeReferenceId { get; set; }

5 references
public string WalletVersion { get; set; }

3 references
public DateTime LastChecked { get; set; }
```