Sending Queries

Learn how to send queries 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, });

```

SecretJS Queries

secretjs.query.auth.account()

Returns account details based on address.

```

Copy const{address,accountNumber,sequence}=awaitsecretjs.query.auth.account({ address:myAddress, });

```

secretjs.query.auth.accounts()

Returns all existing accounts on the blockchain.

```

Copy /// Get all accounts constresult=awaitsecretjs.query.auth.accounts({});

```

secretjs.query.auth.params()

Queries all x/auth parameters.

```

Copy const{ params: { maxMemoCharacters, sigVerifyCostEd25519, sigVerifyCostSecp256k1, txSigLimit, txSizeCostPerByte, }, }=awaitsecretjs.query.auth.params();

```

secretjs.query.authz.grants()

Returns list of authorizations, granted to the grantee by the granter.

secretjs.query.bank.balance()

Balance queries the balance of a single coin for a single account.

```

Copy const{balance}=awaitsecretjs.query.bank.balance({ address:myAddress, denom:"uscrt", });

```

secretjs.query.bank.allBalances()

AllBalances queries the balance of all coins for a single account.

secretjs.query.bank.totalSupply()

TotalSupply queries the total supply of all coins.

secretjs.query.bank.supplyOf()

SupplyOf queries the supply of a single coin.

secretjs.query.bank.params()

Params queries the parameters of x/bank module.

secretjs.query.bank.denomMetadata()

DenomsMetadata queries the client metadata of a given coin denomination.

secretjs.query.bank.denomsMetadata()

DenomsMetadata queries the client metadata for all registered coin denominations.

secretjs.query.compute.codeHashByContractAddress()

Get codeHash of a Secret Contract.

secretjs.query.compute.codeHashByCodeId()

Get codeHash from a code id.

secretjs.query.compute.contractInfo()

Get metadata of a Secret Contract.

secretjs.query.compute.contractsByCode()

Get all contracts that were instantiated from a code id.

secretjs.query.compute.queryContract()

Query a Secret Contract

```

Copy typeResult={ token_info:{ decimals:number; name:string; symbol:string; total_supply:string; }; };

constresult=(awaitsecretjs.query.compute.queryContract({ contract_address:sScrtAddress, code_hash:sScrtCodeHash,// optional but way faster query:{ token_info:{} }, }))asResult;

```

secretjs.query.compute.code()

Get WASM bytecode and metadata for a code id.

```

Copy const{codeInfo}=awaitsecretjs.query.compute.code(codeId);

```

secretjs.query.compute.codes()

Query all contract codes on-chain.

secretjs.query.compute.contractHistory()

Get upgrades history of a Secret Contract.

secretjs.query.distribution.params()

Params queries params of the distribution module.

secretjs.query.distribution.validatorOutstandingRewards()

ValidatorOutstandingRewards queries rewards of a validator address.

secretjs.query.distribution.validatorCommission()

ValidatorCommission queries accumulated commission for a validator.

secretjs.query.distribution.validatorSlashes()

ValidatorSlashes queries slash events of a validator.

secretjs.query.distribution.delegationRewards()

DelegationRewards queries the total rewards accrued by a delegation.

secretjs.query.distribution.delegationTotalRewards()

DelegationTotalRewards queries the total rewards accrued by a each validator.

secretjs.query.distribution.delegatorValidators()

DelegatorValidators queries the validators of a delegator.

secretjs.query.distribution.delegatorWithdrawAddress()

DelegatorWithdrawAddress queries withdraw address of a delegator.

secretjs.query.distribution.communityPool()

CommunityPool queries the community pool coins.

secretjs.query.distribution.foundationTax()

DelegatorWithdrawAddress queries withdraw address of a delegator.

secretjs.query.evidence.evidence()

Evidence queries evidence based on evidence hash.

secretjs.query.evidence.allEvidence()

AllEvidence queries all evidence.

secretjs.query.feegrant.allowance()

Allowance returns fee granted to the grantee by the granter.

secretjs.query.feegrant.allowances()

Allowances returns all the grants for address.

secretjs.query.gov.proposal()

Proposal queries proposal details based on ProposalID.

secretjs.query.gov.proposals()

Proposals queries all proposals based on given status.

```

Copy // Get all proposals const{proposals}=awaitsecretjs.query.gov.proposals({
proposal_status:ProposalStatus.PROPOSAL_STATUS_UNSPECIFIED, voter:"", depositor:"", });

```

secretjs.query.gov.vote()

Vote queries voted information based on proposalID, voterAddr.

secretjs.query.gov.votes()

Votes queries votes of a given proposal.

secretjs.query.gov.params()

Params queries all parameters of the gov module.

secretjs.query.gov.deposit()

Deposit queries single deposit information based proposalID, depositAddr.

```

Copy const{ deposit: {amount}, }=awaitsecretjs.query.gov.deposit({ depositor:myAddress, proposalId:propId, });

```

secretjs.query.gov.deposits()

Deposits queries all deposits of a single proposal.

secretjs.query.gov.tallyResult()

TallyResult queries the tally of a proposal vote.

secretjs.query.ibc_channel.channel()

Channel queries an IBC Channel.

secretjs.query.ibc_channel.channels()

Channels queries all the IBC channels of a chain.

secretjs.query.ibc_channel.connectionChannels()

ConnectionChannels queries all the channels associated with a connection end.

secretjs.query.ibc_channel.channelClientState()

ChannelClientState queries for the client state for the channel associated with the provided channel identifiers.

secretjs.query.ibc_channel.channelConsensusState()

ChannelConsensusState queries for the consensus state for the channel associated with the provided channel identifiers.

secretjs.query.ibc_channel.packetCommitment()

PacketCommitment queries a stored packet commitment hash.

secretjs.query.ibc_channel.packetCommitments()

PacketCommitments returns all the packet commitments hashes associated with a channel.

secretjs.query.ibc_channel.packetReceipt()

PacketReceipt queries if a given packet sequence has been received on the queried chain

secretjs.query.ibc_channel.packetAcknowledgement()

PacketAcknowledgement queries a stored packet acknowledgement hash.

secretjs.query.ibc_channel.packetAcknowledgements()

PacketAcknowledgements returns all the packet acknowledgements associated with a channel.

secretjs.query.ibc_channel.unreceivedPackets()

UnreceivedPackets returns all the unreceived IBC packets associated with a channel and sequences.

secretjs.query.ibc_channel.unreceivedAcks()

UnreceivedAcks returns all the unreceived IBC acknowledgements associated with a channel and sequences.

secretjs.query.ibc_channel.nextSequenceReceive()

NextSequenceReceive returns the next receive sequence for a given channel.

secretjs.query.ibc_client.clientState()

ClientState queries an IBC light client.

secretjs.query.ibc_client.clientStates()

ClientStates queries all the IBC light clients of a chain.

secretjs.query.ibc_client.consensusState()

ConsensusState queries a consensus state associated with a client state at a given height.

secretjs.query.ibc_client.consensusStates()

ConsensusStates queries all the consensus state associated with a given client.

secretjs.query.ibc_client.clientStatus()

Status queries the status of an IBC client.

secretjs.query.ibc_client.clientParams()

ClientParams queries all parameters of the ibc client.

secretjs.query.ibc_client.upgradedClientState()

UpgradedClientState queries an Upgraded IBC light client.

secretjs.query.ibc_client.upgradedConsensusState()

UpgradedConsensusState queries an Upgraded IBC consensus state.

secretjs.query.ibc_connection.connection()

Connection queries an IBC connection end.

secretjs.query.ibc_connection.connections()

Connections queries all the IBC connections of a chain.

secretjs.query.ibc_connection.clientConnections()

ClientConnections queries the connection paths associated with a client state.

secretjs.query.ibc_connection.connectionClientState()

ConnectionClientState queries the client state associated with the connection.

secretjs.query.ibc_connection.connectionConsensusState()

ConnectionConsensusState queries the consensus state associated with the connection.

secretjs.query.ibc_transfer.denomTrace()

DenomTrace queries a denomination trace information.

secretjs.query.ibc_transfer.denomTraces()

DenomTraces queries all denomination traces.

secretjs.query.ibc_transfer.params()

Params queries all parameters of the ibc-transfer module.

secretjs.query.mint.params()

Params returns the total set of minting parameters.

secretjs.query.mint.inflation()

Inflation returns the current minting inflation value.

secretjs.query.mint.annualProvisions()

AnnualProvisions current minting annual provisions value.

secretjs.query.params.params()

Params queries a specific parameter of a module, given its subspace and key.

secretjs.query.registration.txKey()

Returns the key used for transactions.

secretjs.query.registration.registrationKey()

Returns the key used for registration.

secretjs.query.registration.encryptedSeed()

Returns the encrypted seed for a registered node by public key.

secretjs.query.slashing.params()

Params queries the parameters of slashing module.

secretjs.query.slashing.signingInfo()

SigningInfo queries the signing info of given cons address.

secretjs.query.slashing.signingInfos()

SigningInfos queries signing info of all validators.

secretjs.query.staking.validators()

Validators queries all validators that match the given status.

```

Copy // Get all validators const{validators}=awaitsecretjs.query.staking.validators({ status:""});

```

secretjs.query.staking.validator()

Validator queries validator info for given validator address.

secretjs.query.staking.validatorDelegations()

ValidatorDelegations queries delegate info for given validator.

secretjs.query.staking.validatorUnbondingDelegations()

ValidatorUnbondingDelegations queries unbonding delegations of a validator.

secretjs.query.staking.delegation()

Delegation queries delegate info for given validator delegator pair.

secretjs.query.staking.unbondingDelegation()

UnbondingDelegation queries unbonding info for given validator delegator pair.

secretjs.query.staking.delegatorDelegations()

DelegatorDelegations queries all delegations of a given delegator address.

secretjs.query.staking.delegatorUnbondingDelegations()

DelegatorUnbondingDelegations queries all unbonding delegations of a given delegator address.

secretjs.query.staking.redelegations()

Redelegations queries redelegations of given address.

secretjs.query.staking.delegatorValidators()

DelegatorValidators queries all validators info for given delegator address.

secretjs.query.staking.delegatorValidator()

DelegatorValidator queries validator info for given delegator validator pair.

secretjs.query.staking.historicalInfo()

HistoricalInfo queries the historical info for given height.

secretjs.query.staking.pool()

Pool queries the pool info.

secretjs.query.staking.params()

Parameters queries the staking parameters.

secretjs.query.tendermint.getNodeInfo()

GetNodeInfo queries the current node info.

secretjs.query.tendermint.getSyncing()

GetSyncing queries node syncing.

secretjs.query.tendermint.getLatestBlock()

GetLatestBlock returns the latest block.

secretjs.query.tendermint.getBlockByHeight()

GetBlockByHeight queries block for given height.

secretjs.query.tendermint.getLatestValidatorSet()

GetLatestValidatorSet queries latest validator-set.

secretjs.query.tendermint.getValidatorSetByHeight()

GetValidatorSetByHeight queries validator-set at a given height.

secretjs.query.upgrade.currentPlan()

CurrentPlan queries the current upgrade plan.

secretjs.query.upgrade.appliedPlan()

AppliedPlan queries a previously applied upgrade plan by its name.

secretjs.query.upgrade.upgradedConsensusState()

UpgradedConsensusState queries the consensus state that will serve as a trusted kernel for the next version of this chain. It will only be stored at the last height of this chain.

secretjs.query.upgrade.moduleVersions()

ModuleVersions queries the list of module versions from state.

Last updated3 months ago On this page *Secret Network Client Setup * SecretJS Queries

Was this helpful? Edit on GitHub Export as PDF