

# Build Transactions

## [POST]

/v1/:network/tx-builder

This API returns the SignDoc data for broadcasting transactions. The response includes tx for generating the signature, original messages used to create the SignDoc, fee value, and basic information of the account. Users can either generate the signature directly without 0x from the value and broadcast it, or use the provided broadcast API to send the tx.

Multi sign account is not supported yet.

### Request

Headers Authorization :

Bearer {access\_token} Content-Type :

application/json

### Response

Example { // SignDoc for generating the signature. Use the data without '0x' "tx" :

"0x0aa6010a98010a232f636f736d6f732e7374616b696e612e763162657461312e4d736744656c656761746512710a2d636f736d6f73316772306533706a3379366671767a79666d3071787977396835647", // Original data that user transfer "messages" : [ { "type" :

"cosmos.staking.v1beta1.MsgDelegate" , "delegatorAddress" :

"osmo1gr0e3pj3y6fqvzyam0qxyw9h5dwfrvh8zv3x9p" , "validatorAddress" :

"osmovaloper1clpqr4nrc4khgkxj78fcwwh6dl3uw4epsluffn" , "amount" : { "denom" :

"uosmo" ,

"amount" :

"1" } } ] , // Transaction fee "fee" : { "amount" : [ { "denom" :

"uosmo" ,

"amount" :

"760" } } ,

"gas" :

"303815" } , // Account public key of signer "publicKey" : { "@type" :

"/cosmos.crypto.secp256k1.PubKey" , "key" :

"A00Jvfv1IuvAODaiOPIMZpDTWSq7qvoFV8k7ptsDqLU" } , // Account number of signer "accountNumber" :

"63277" , // Account sequence of signer "sequence" :

"12" }

### How to make a public key from private key

Make Public Key const

privateKey

=

Buffer .from ( '1acbcaf3fec7fcb93afa3cafd8bacf8b1fbcaccc0accdb9ef896fd8eafc135a' , 'hex' , ); const

pubKeyUint8Array

=

TinySecp256k1 .pointFromScalar (privateKey ,

true ); if ( ! pubKeyUint8Array ) { throw

new

Error ( 'Invalid private key' ); }

// Use this publicKey when sending a transaction (tx) for the first time. const

publicKey

=

Buffer .from (pubKeyUint8Array) .toString ( 'base64' );

## Send Transaction

Body { // Signer Address "signer" :

"string" , // Tx Memo "memo" :

"string" , // !!! When sending a transaction (tx) for the first time, must provide the public key // pubkey type ("tendermint/PubKeySecp256k1" | "ethermint/PubKeyEthSecp256k1" | "injective/PubKeyEthSecp256k1") "pubkey" : [ { "type" :

"string" ,

"key" :

"string" } } ] , // Message Array "messages" : [ { // Message Type "type" :

"cosmos.bank.v1beta1.MsgSend" , // Sender Address "fromAddress" :

"string" , // Receiver Address "toAddress" :

"string" , // Amount Array "amount" : [ { "denom" :

"string" , "amount": string } } ] }

### Try API

Method :POST URL /v1/:network/tx-builder CallURL https://apis.mintscan.io/v1/:network/tx-builder Header Bear TokenParameters network Body { " signer " : string "osmo1clpqr4nrc4khgkxj78fcwwh6dl3uw4epasmvnj" " memo " : string "Send Memo" " pubkey " : [ 0 : { " type " : string "tendermint/PubKeySecp256k1" " key " : string "An0YrOygZ23oiJQZg63gJSbg4nkrWmHBC02/6Am2oDrS" } ] " messages " : [ 0 : { " type " : string "cosmos.bank.v1beta1.MsgSend" " fromAddress " : string

"osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnrj" " toAddress " : string "osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnrj" " amount " : [ 0 : { " denom " : string "uosmo" " amount " : string "1" } ] ] } }  
Execute Clear

## Delegate Transaction

Body { // Signer Address "signer" :

"string" , // Tx Memo "memo" :

"string" , // !!! When sending a transaction (tx) for the first time, must provide the public key // pubkey type ("tendermint/PubKeySecp256k1" | "ethermint/PubKeyEthSecp256k1" | "injective/PubKeyEthSecp256k1") "pubkey" : [ { "type" :

"string" ,

"key" :

"string" ] } , // Message Array "messages" : [ { // Message Type "type" :

"cosmos.staking.v1beta1.MsgDelegate" , // Sender Address "delegatorAddress" :

"string" , // Validator Address to delegate "validatorAddress" :

"string" , // Amount "amount" : { "denom" :

"string" , "amount" : string } } ] } } }

### Try API

Method :POST URL /v1:/network/tx-builder CallURL https://apis.mintscan.io/v1:/network/tx-builder Header Bear TokenParameters network Body { " signer " : string "osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnrj" " memo " : string "Delegate Memo" " pubkey " : [ 0 : { " type " : string "tendermint/PubKeySecp256k1" " key " : string "AnOyrOygz23oiJQZg63gJSbg4nrkWmHBC02/6Am2oDrS" } ] " messages " : [ 0 : { " type " : string "cosmos.staking.v1beta1.MsgDelegate" " delegatorAddress " : string "osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnrj" " validatorAddress " : string "osmovaloper1clpqr4nrk4khgkxj78fcwwh6dl3uw4ep88n0y4" " amount " : { " denom " : string "uosmo" " amount " : string "1" } } ] } } Execute Clear

## Undelegate Transaction

Body { // Signer Address "signer" :

"string" , // Tx Memo "memo" :

"string" , // !!! When sending a transaction (tx) for the first time, must provide the public key // pubkey type ("tendermint/PubKeySecp256k1" | "ethermint/PubKeyEthSecp256k1" | "injective/PubKeyEthSecp256k1") "pubkey" : [ { "type" :

"string" ,

"key" :

"string" ] } , // Message Array "messages" : [ { // Message Type "type" :

"cosmos.staking.v1beta1.MsgUndelegate" , // Sender Address "delegatorAddress" :

"string" , // Validator Address to undelegate "validatorAddress" :

"string" , // Amount "amount" : { "denom" :

"string" , "amount" : string } } ] } }

### Try API

Method :POST URL /v1:/network/tx-builder CallURL https://apis.mintscan.io/v1:/network/tx-builder Header Bear TokenParameters network Body { " signer " : string "osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnrj" " memo " : string "Undelegate Memo" " pubkey " : [ 0 : { " type " : string "tendermint/PubKeySecp256k1" " key " : string "AnOyrOygz23oiJQZg63gJSbg4nrkWmHBC02/6Am2oDrS" } ] " messages " : [ 0 : { " type " : string "cosmos.staking.v1beta1.MsgUndelegate" " delegatorAddress " : string "osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnrj" " validatorAddress " : string "osmovaloper1clpqr4nrk4khgkxj78fcwwh6dl3uw4ep88n0y4" " amount " : { " denom " : string "uosmo" " amount " : string "1" } } ] } } Execute Clear

## Begin Redelegate Transaction

Body { // Signer Address "signer" :

"string" , // Tx Memo "memo" :

"string" , // !!! When sending a transaction (tx) for the first time, must provide the public key // pubkey type ("tendermint/PubKeySecp256k1" | "ethermint/PubKeyEthSecp256k1" | "injective/PubKeyEthSecp256k1") "pubkey" : [ { "type" :

"string" ,

"key" :

"string" ] } , // Message Array "messages" : [ { // Message Type "type" :

"cosmos.staking.v1beta1.MsgBeginRedelegate" , // Sender Address "delegatorAddress" :

"string" , // From Validator Address to redelegate "validatorSrcAddress" :

"string" , // To Validator Address to redelegate "validatorDstAddress" :

"string" , // Amount "amount" : { "denom" :

"string" , "amount" : string } } ] } }

### Try API

Method :POST URL /v1:/network/tx-builder CallURL https://apis.mintscan.io/v1:/network/tx-builder Header Bear TokenParameters network Body { " signer " : string "osmo1des8hpaqnxgdjahelvhgeoctpnq7gm04344cdl" " memo " : string "Begin Redelegate Memo" " pubkey " : [ 0 : { " type " : string "tendermint/PubKeySecp256k1" " key " : string "AnOyrOygz23oiJQZg63gJSbg4nrkWmHBC02/6Am2oDrS" } ] " messages " : [ 0 : { " type " : string "cosmos.staking.v1beta1.MsgBeginRedelegate" " delegatorAddress " : string "osmo1des8hpaqnxgdjahelvhgeoctpnq7gm04344cdl" " validatorSrcAddress " : string "osmovaloper14lztv4gdwh2q4ymyjqma0p4j4aykpn92l4warr" " validatorDstAddress " : string "osmovaloper1clpqr4nrk4khgkxj78fcwwh6dl3uw4ep88n0y4" " amount " : { " denom " : string "uosmo" " amount " : string "1" } } ] } } Execute Clear

## Claim Delegate Rewards Transaction

Body { // Signer Address "signer" :

"string" , // Tx Memo "memo" :

"string" , // !!! When sending a transaction (tx) for the first time, must provide the public key // pubkey type ("tendermint/PubKeySecp256k1" | "ethermint/PubKeyEthSecp256k1" | "injective/PubKeyEthSecp256k1") "pubkey" : [ { "type" :

"string" ,

"key" :

"string" ] } , // Message Array "messages" : [ { // Message Type "type" :

"cosmos.distribution.v1beta1.MsgWithdrawDelegatorReward" , // Sender Address "delegatorAddress" :

"string" , // Validator Address to claim rewards "validatorAddress"": string" } } }

### Try API

Method :POST URL /v1/:network/tx-builder CallURL https://apis.mintscan.io/v1/:network/tx-builder Header Bear Token *Parameters network* Body { " signer " : string "osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnj" " memo " : string "Withdraw Delegate Reward Memo" " pubkey " : [ 0 : { " type " : string "tendermint/PubKeySecp256k1" " key " : string "An0YrOygZ23oiJQZg63gJSbg4nrWmHBC02/6Am2oDrS" } ] " messages " : [ 0 : { " type " : string "cosmos.distribution.v1beta1.MsgWithdrawDelegatorReward" " delegatorAddress " : string "osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnj" " validatorAddress " : string "osmovaloper1clpqr4nrk4khgkxj78fcwwh6dl3uw4ep88n0y4" } ] } Execute Clear

## Claim Validator Commissions Transaction

Body { // Signer Address "signer" :

"string" , // Tx Memo "memo" :

"string" , // !!! When sending a transaction (tx) for the first time, must provide the public key // pubkey type ("tendermint/PubKeySecp256k1" | "ethermint/PubKeyEthSecp256k1" | "injective/PubKeyEthSecp256k1") "pubkey" : { { "type" :

"string" ,

"key" :

"string" } } , // Message Array "messages" : [ { // Message Type "type" :

"cosmos.distribution.v1beta1.MsgWithdrawValidatorCommission" , // Validator Address to withdraw validator commission "validatorAddress"": string" } } }

### Try API

Method :POST URL /v1/:network/tx-builder CallURL https://apis.mintscan.io/v1/:network/tx-builder Header Bear Token *Parameters network* Body { " signer " : string "osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnj" " memo " : string "Withdraw Validator Commission Memo" " pubkey " : [ 0 : { " type " : string "tendermint/PubKeySecp256k1" " key " : string "An0YrOygZ23oiJQZg63gJSbg4nrWmHBC02/6Am2oDrS" } ] " messages " : [ 0 : { " type " : string "cosmos.distribution.v1beta1.MsgWithdrawValidatorCommission" " validatorAddress " : string "osmovaloper1clpqr4nrk4khgkxj78fcwwh6dl3uw4ep88n0y4" } ] } Execute Clear

## Governance Vote Transaction

Body { // Signer Address "signer" :

"string" , // Tx Memo "memo" :

"string" , // !!! When sending a transaction (tx) for the first time, must provide the public key // pubkey type ("tendermint/PubKeySecp256k1" | "ethermint/PubKeyEthSecp256k1" | "injective/PubKeyEthSecp256k1") "pubkey" : { { "type" :

"string" ,

"key" :

"string" } } , // Message Array "messages" : [ { // Message Type "type" :

"cosmos.gov.v1beta1.MsgVote" , // Proposal ID "proposald" :

"string" , // Voter "voter" :

"string" , // Vote Option("VOTE\_OPTION\_UNSPECIFIED" | "VOTE\_OPTION\_YES" | "VOTE\_OPTION\_ABSTAIN" | "VOTE\_OPTION\_NO" | "VOTE\_OPTION\_NO\_WITH\_VETO") "option"": string" } } }

### Try API

Method :POST URL /v1/:network/tx-builder CallURL https://apis.mintscan.io/v1/:network/tx-builder Header Bear Token *Parameters network* Body { " signer " : string "osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnj" " memo " : string "Governance Vote Memo" " pubkey " : [ 0 : { " type " : string "tendermint/PubKeySecp256k1" " key " : string "An0YrOygZ23oiJQZg63gJSbg4nrWmHBC02/6Am2oDrS" } ] " messages " : [ 0 : { " type " : string "cosmos.gov.v1beta1.MsgVote" " proposald " : string "565" " voter " : string "osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnj" " option " : string "VOTE\_OPTION\_YES" } ] } Execute Clear

## Wasm Contract Execute Transaction

Body { // Signer Address "signer" :

"string" , // Tx Memo "memo" :

"string" , // !!! When sending a transaction (tx) for the first time, must provide the public key // pubkey type ("tendermint/PubKeySecp256k1" | "ethermint/PubKeyEthSecp256k1" | "injective/PubKeyEthSecp256k1") "pubkey" : { { "type" :

"string" ,

"key" :

"string" } } , // Message Array "messages" : [ { // Message Type "type" :

"cosmwasm.wasm.v1.MsgExecuteContract" , // Sender "sender" :

"string" , // Contract Address "contract" :

"string" , // Execute Messages "msg" : { } , // Funds "funds" : [ ] } } }

### Try API

Method :POST URL /v1/:network/tx-builder CallURL https://apis.mintscan.io/v1/:network/tx-builder Header Bear Token *Parameters network* Body { " signer " : string "neutron1aaaaaaaaah3rrtmhmp5aymz808mn2vpqgu0l" " memo " : string "Wasm Contract Execute Memo" " pubkey " : [ 0 : { " type " : string "tendermint/PubKeySecp256k1" " key " : string "An0YrOygZ23oiJQZg63gJSbg4nrWmHBC02/6Am2oDrS" } ] " messages " : [ 0 : { " type " : string "cosmwasm.wasm.v1.MsgExecuteContract" " sender " : string "neutron1aaaaaaaaah3rrtmhmp5aymz808mn2vpqgu0l" " contract " : string "neutron1kwp7pd4ts6gukm3e820kyftz4vv5jqtmal8pwqezrnq2ddycqasr87x9p" " msg " : { " claim " : { } } " funds " : [ ] ] } Execute Clear

## Authz Grant Transaction

Body { // Signer Address "signer" :

"string" , // Tx Memo "memo" :

"string" , // !!! When sending a transaction (tx) for the first time, must provide the public key // pubkey type ("tendermint/PubKeySecp256k1" | "ethermint/PubKeyEthSecp256k1" | "injective/PubKeyEthSecp256k1") "pubkey" : { { "type" :

"string" ,

"key" :

"string" } } , // Message Array "messages" : [ { // GeneraicAuthorization "type" :

"cosmos.authz.v1beta1.MsgGrant" , "granter" :

"string" , "grantee" :

"string" , "grant" : { " authorization " : { // Auth Type("GenericAuthorization" | "StakeAuthorization" | "SendAuthorization") "type" :

```
"GenericAuthorization" , "msg" :

"string" } , "expiration" :

"datetime" } } , { // StakeAuthorization "type" :

"cosmos.authz.v1beta1.MsgGrant" , "granter" :

"string" , "grantee" :

"string" , "grant" : { "authorization" : { // Auth Type("GenericAuthorization" | "StakeAuthorization" | "SendAuthorization") "type" :

"StakeAuthorization" , "maxTokens" : { "denom" :

"string" ,

"amount" :

"string" } , // When used in conjunction with a denyList, the allowList is not applied "allowList" : { "address" : [ "string" ] } , // Authorization
Type("AUTHORIZATION_TYPE_DELEGATE" | "AUTHORIZATION_TYPE_UNDELEGATE" | "AUTHORIZATION_TYPE_REDELEGATE") "authorizationType" :

"string" } , "expiration" :

"datetime" } } , { // SendAuthorization "type" :

"cosmos.authz.v1beta1.MsgGrant" , "granter" :

"string" , "grantee" :

"string" , "grant" : { "authorization" : { // Auth Type("GenericAuthorization" | "StakeAuthorization" | "SendAuthorization") "type" :

"SendAuthorization" , "spendLimit" : { "denom" :

"string" ,

"amount" :

"string" }}} , "expiration" :

"datetime" } } } }
```

## Try API

Method :POST URL /v1:/network/tx-builder CallURL <https://apis.mintscan.io/v1:/network/tx-builder> Header Bear Token *Parameters network* Body { " signer " : string  
"osmo1aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" " memo " : string "Authz Grant Memo" " pubkey " : [ 0 : { " type " : string "tendermint/PubKeySecp256k1" " key " : string  
"An0yrOyg223oiJQZg63gJSbg4nkrWmHBC02/6Am2oDrS" } ] " messages " : [ 0 : { " type " : string "cosmos.authz.v1beta1.MsgGrant" " granter " : string  
"osmo1aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" " grantee " : string "osmo1bb" " grant " : { " authorization " : { " type " : string  
"GenericAuthorization" " msg " : string "/cosmos.gov.v1beta1.MsgVote" } " expiration " : string "2023-12-31 00:00:00" } } ] Execute Clear

## Authz Execute Transaction

Body { // Signer Address "signer" :  
  
"string" , // Tx Memo "memo" :  
  
"string" , // !!! When sending a transaction (tx) for the first time, must provide the public key // pubkey type ("tendermint/PubKeySecp256k1" | "ethermint/PubKeyEthSecp256k1" |  
"injective/PubKeyEthSecp256k1") "pubkey" : [{ "type" :  
  
"string" ,  
  
"key" :  
  
"string" } ] , // Message Array "messages" : [ { "type" :  
  
"cosmos.authz.v1beta1.MsgExec" , "grantee" :  
  
"string" , "msgs" : [ { // type\_url excluding '/' (ex: cosmos.bank.v1beta1.MsgSend) "type" :  
  
"string" // the values included in the corresponding type message } } ] }

## Try API

Method :POST URL /v1:/network/tx-builder CallURL <https://apis.mintscan.io/v1:/network/tx-builder> Header Bear Token *Parameters network* Body { " signer " : string  
"osmo1bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb" " memo " : string "Authz Exec Memo" " pubkey " : [ 0 : { " type " : string "tendermint/PubKeySecp256k1" " key " : string  
"An0yrOyg223oiJQZg63gJSbg4nkrWmHBC02/6Am2oDrS" } ] " messages " : [ 0 : { " type " : string "cosmos.authz.v1beta1.MsgExec" " grantee " : string  
"osmo1bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb" " msgs " : [ 0 : { " type " : string "cosmos.bank.v1beta1.MsgSend" " fromAddress " : string  
"osmo1aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" " toAddress " : string "osmo1cccccccccccccccccccccccccccccccccccccc" " amount " : [ 0 : { " denom " : string "uosmo" " amount " : string "1" } ] }  
] } ] Execute Clear

## Authz Revoke Transaction

Body { // Signer Address "signer" :  
  
"string" , // Tx Memo "memo" :  
  
"string" , // !!! When sending a transaction (tx) for the first time, must provide the public key // pubkey type ("tendermint/PubKeySecp256k1" | "ethermint/PubKeyEthSecp256k1" |  
"injective/PubKeyEthSecp256k1") "pubkey" : [{ "type" :  
  
"string" ,  
  
"key" :  
  
"string" } ] , // Message Array "messages" : [ { "type" :  
  
"cosmos.authz.v1beta1.MsgRevoke" , "granter" :  
  
"string" , "grantee" :  
  
"string" , // msgTypeUrl // GenericAuthorization, you can simply use the msg value of that authorization as it is // SendAuthorization, you should use /cosmos.bank.v1beta1.MsgSend //  
StakeAuthorization, you should provide values based on the Authorization Type granted. // - AUTHORIZATION\_TYPE\_DELEGATE, use /cosmos.staking.v1beta1.MsgDelegate. // -  
AUTHORIZATION\_TYPE\_UNDELEGATE, use /cosmos.staking.v1beta1.MsgUndelegate // - AUTHORIZATION\_TYPE\_REDELEGATE, use /cosmos.staking.v1beta1.MsgBeginRedelegate  
"msgTypeUrl" :  
  
"string" } ] }

## Try API

Method :POST URL /v1:/network/tx-builder CallURL <https://apis.mintscan.io/v1:/network/tx-builder> Header Bear Token *Parameters network* Body { " signer " : string  
"osmo1aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" " memo " : string "Authz Revoke Memo" " pubkey " : [ 0 : { " type " : string "tendermint/PubKeySecp256k1" " key " : string  
"An0yrOyg223oiJQZg63gJSbg4nkrWmHBC02/6Am2oDrS" } ] " messages " : [ 0 : { " type " : string "cosmos.authz.v1beta1.MsgRevoke" " granter " : string  
"osmo1aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" " grantee " : string "osmo1bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb" " msgTypeUrl " : string "/cosmos.bank.v1beta1.MsgSend" } } ]  
Execute Clear

## Multiple Message Type of Transaction

Body { // Signer Address "signer" :

"string" , // Tx Memo "memo" :

"string" , // !!! When sending a transaction (tx) for the first time, must provide the public key // pubkey type ("tendermint/PubKeySecp256k1" | "ethermint/PubKeyEthSecp256k1" | "injective/PubKeyEthSecp256k1") "pubkey" : [{ "type" :

"string" ,

"key" :

"string" ]} , // Message Array "messages" : [ // The message types within the messages defined above ] }

### Try API

Method :POST URL /v1/:network/tx-builder CallURL https://apis.mintscan.io/v1/:network/tx-builder Header Bear Token*Parameters network* Body { " signer " : string  
"osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnj" " memo " : string "Withdraw Rewards And Commissions Memo" " pubkey " : [ 0 : { " type " : string "tendermint/PubKeySecp256k1" " key " : string  
"An0yrOygz23oijQZg63gJSbg4nkrWmHBC02/6Am2oDrS" } ] " messages " : [ 0 : { " type " : string "cosmos.distribution.v1beta1.MsgWithdrawDelegatorReward" " delegatorAddress " : string  
"osmo1clpqr4nrk4khgkxj78fcwwh6dl3uw4epasmvnj" " validatorAddress " : string "osmovaloper1clpqr4nrk4khgkxj78fcwwh6dl3uw4ep88n0y4" } 1 : { " type " : string  
"cosmos.distribution.v1beta1.MsgWithdrawValidatorCommission" " validatorAddress " : string "osmovaloper1clpqr4nrk4khgkxj78fcwwh6dl3uw4ep88n0y4" } ] } Execute Clear Last updated onSeptember  
6, 2023 [Transactions Hash](#) [Sign Transaction](#)