How Make Signature

This documentation page provides an example of generating a signature using the tx (SignDoc) produced in theBuild Transaction process. The example code in this document is written in Node.js. Users can implement the sign logic based on their respective programming language.

Make Signature // Import packages for sign import

*

as TinySecp256k1 from

'tiny-secp256k1' ; import

*

as encHex from

'crypto-js/enc-hex' ; import

*

as sha256 from

'crypto-js/sha256' ;

// Define a custom function to execute the logic described on the Build Transaction page. // Each user can implement their own version of this function and use it accordingly const

buildResult

=

await

buildTx (); / *Example of result { "tx": "0x0aa6010a98010a232f636f736d6f732e7374616b696e612e763162657461312e4d736744656c656761746512710a2d636f736d6f73316772306533706a3379366671767a79666d3071787977396835647 "messages": [{ "type": "cosmos.staking.v1beta1.MsgDelegate", "delegatorAddress": "osmo1gr0e3pj3y6fqvzyam0qxyw9h5dwfrvh8zv3x9p", "validatorAddress": "osmovaloper1clpqr4nrc4khgkxj78fcwwh6dl3uw4epsluffn", "amount": { "denom": "uosmo", "amount": "1" } }], "fee": { "amount": [{ "denom": "uosmo", "amount": "760" }], "gas": "303815" }, "publicKey": { "@type": "/cosmos.crypto.secp256k1.PubKey", "key": "A00Jvfv1luvAODaiOPIMZpDTWSq7qvoFV8k7ptdsDqLU" }, "accountNumber": "63277", "sequence": "12" }* /

// Removes the '0x' string from the beginning of the tx. const

signDoc

=

buildResult . tx .replace (/ ^ 0x(. *)/ ,

'1'); // SignDoc Example: 0aa6010a98010a232f636f736d6f732e7374616b696e612e763162657461312e4d736744656c656761746512710a2d636f736d6f73316772306533706a3379366671767a79666d307178797977396835647764776

// The part that creates a Buffer using the user's Private Key. // If the Private Key starts with '0x', the '0x' prefix is removed and the value is used. const

privateKey

=

Buffer .from ('ca7e19f112f2a01aaccefba10840bdcbcc16344c310afff65397eaa940e51f1d' , 'hex' ,); const

hashedSignDoc

=

sha256 (encHex .parse (signDoc)) .toString (encHex); const

signResult

=

TinySecp256k1 .sign (Buffer .from (hashedSignDoc ,

'hex') , privateKey);

const

signatureHex

=

Buffer .from (signResult) .toString ('hex');

// Use this to broadcast the transaction const

signature

=

0x { signatureHex } ; // Example: 0x10d0688f545e01802023a12cba8a69be50bd6e504b21781d97e7baaab61c4e8619753e64a377e85bf1488b4cdc34c87ce49c4d0d8fc0b52857129bf5aeccaed5 Last updated onAugust 30, 2023 Build Transaction Broadcast Transaction