# Signing a Message

Suggest Edits

When a web application has established a connection to the ME wallet, it can prompt users to sign a message. This is commonplace amongst many dApps, as it gives application owners the ability to verify ownership of the wallet. Signing a message does not require any transaction fees.

Sats Connect provides an easy way to request message signing. Let's take a look at the code below.

JavaScript const connectionStatus = useContext(ConnectionStatusContext); const nativeSegwitAddress = connectionStatus?.accounts[1]?.address;

async function signWalletMessage() { try { await signMessage({ payload: { network: { type: BitcoinNetworkType.Mainnet, }, address: nativeSegwitAddress, message: 'Hello World. Welcome to the Magic Eden wallet!', }, onFinish: (response) => { alert(Successfully signed message: {response}); }, onCancel: () => { alert('Request canceled'); }, }); } catch (err) { console.error(err); } } This piece of code makes use of two import from Sats Connect. BitcoinNetworkType, which is a basic enum to pick the network, and signMessage. The latter takes care of a lot of the core functionality of message signing, leaving us to provide basic parameters.

import { BitcoinNetworkType, signMessage } from 'sats-connect'; A successful prompt to sign a message will look something like this: Updatedabout 1 month ago