

Signing a Message

[Suggest Edits](#)

Once a web application has connected to the ME Solana wallet, it can prompt users to sign a message, send transactions, and more. Signing 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.

Let's take a look at what this looks like:

```
JavaScript import { MagicEdenProvider } from "../types/types";
```

```
/* * Prompts the user to sign a message using the MagicEden Provider object * @param {MagicEdenProvider} provider The MagicEdenProvider object * @param {string} message The message to sign * @returns {Promise} A promise that resolves to the signed message * @throws {Error} If an unexpected error occurs while signing the message / export const signMessage = async ( provider: MagicEdenProvider, message: string ): Promise => { try { const encodedText = new TextEncoder().encode(message); const signedMessage = await provider.signMessage(encodedText); return signedMessage; } catch (error) { console.warn(error); throw new Error("An unexpected error occurred while signing the message."); } }; Signing a message is relatively simple and doesn't require any real interaction with web3 libraries. Check out the signMessage functionality in the demo app for a more well rounded implementation. Updated about 1 month ago
```