Ethereum provider API

This page is a reference for the Ethereum provider API of MetaMask'sWallet API . MetaMask injects the provider API into websites visited by its users using thewindow.ethereum provider object. You can use the providerproperties ,methods , andevents in your dapp.

Note MetaMask supportsEIP-6963 , which introduces an alternative wallet detection mechanism to thewindow.ethereum injected provider. This alternative mechanism enables dapps to supportwallet interoperability by discovering multiple injected wallet providers in a user's browser. We recommendusing this mechanism to connect to MetaMask.

You can access the provider API using the selected EIP-6963 provider object. Throughout this documentation, we refer to the selected provider usingprovider .

Properties

isMetaMask

This property istrue if the user has MetaMask installed, andfalse otherwise.

note This property is non-standard. Non-MetaMask providers may also set this property totrue .

Example

provider . isMetaMask // Or window.ethereum.isMetaMask if you don't support EIP-6963.

Methods

isConnected()

Indicates whether the provider is connected to the current chain. If the provider isn't connected, the page must be reloaded to re-establish the connection. See theconnect anddisconnect events for more information.

note This method is unrelated toaccessing a user's accounts . In the provider interface, "connected" and "disconnected" refer to whether the provider can make RPC requests to the current chain.

Parameters

None.

Returns

true if the provider is connected to the current chain,false otherwise.

Example

provider . isConnected ( )

// Or window.ethereum.isConnected() if you don't support EIP-6963.

request()

This method is used to submitJSON-RPC API requests to Ethereum using MetaMask.

Parameters

An object containing:

- method
- :string
-

    - The JSON-RPC API method name.
- params
- :array
- orobject
-
    - (Optional) Parameters of the RPC method.
- In practice, if a method has parameters, they're almost always of typearray
- .

Returns

A promise that resolves to the result of the RPC method call. If the request fails, the promise rejects with an error .

Example

The following is an example of usingrequest() to call eth_sendTransaction :

provider // Or window.ethereum if you don't support EIP-6963. . request ({ method :

"eth_sendTransaction" , params :

[{ from :

"0xb60e8dd61c5d32be8058bb8eb970870f07233155" , to :

"0xd46e8dd67c5d32be8058bb8eb970870f07244567" , gas :

"0x76c0" ,

// 30400 gasPrice :

"0x9184e72a000" ,

// 10000000000000 value :

"0x9184e72a" ,

// 2441406250 data :

"0xd46e8dd67c5d32be8d46e8dd67c5d32be8058bb8eb970870f072445675058bb8eb970870f072445675" , } ,] , }) . then ((result)

=>

{ // The result varies by RPC method. // For example, this method returns a transaction hash hexadecimal string upon success. }) . catch ((error)

=>

{ // If the request fails, the Promise rejects with an error. })

_metamask.isUnlocked()

caution This method is experimental. Use it at your own risk. Indicates if MetaMask is unlocked by the user. MetaMask must be unlocked to perform any operation involving user accounts. Note that this method doesn't indicate if the user has exposed any accounts to the caller.

Parameters

None.

Returns

A promise that resolves totrue if MetaMask is unlocked by the user, andfalse otherwise.

Example

provider . _metamask . isUnlocked ( )

// Or window.ethereum._metamask.isUnlocked() if you don't support EIP-6963.

Events

The MetaMask provider emits events using the Node.jsEventEmitter API. The following is an example of listening to theaccountsChanged event.

You shouldremove listeners after you're done listening to an event (for example, on componentunmount in React).

function

handleAccountsChanged (accounts)

{ // Handle new accounts, or lack thereof. }

provider // Or window.ethereum if you don't support EIP-6963. . on ("accountsChanged" , handleAccountsChanged)

// Later

provider // Or window.ethereum if you don't support EIP-6963. . removeListener ("accountsChanged" ,
handleAccountsChanged)

accountsChanged

provider // Or window.ethereum if you don't support EIP-6963. . on ("accountsChanged" ,

handler :

(accounts :

Array < string

)

=>

void) ; The provider emits this event when the return value of theeth_accounts RPC method changes.eth_accounts returns either an empty array, or an array that contains the addresses of the accounts the caller is permitted to access with the most recently used account first. Callers are identified by their URL origin, which means that all sites with the same origin share the same permissions.

This means that the provider emitsaccountsChanged when the user's exposed account address changes. Listen to this event tohandle accounts .

chainChanged

provider // Or window.ethereum if you don't support EIP-6963. . on ("chainChanged" ,

handler :

(chainId :

string)

=>

void) ; The provider emits this event when the currently connected chain changes. Listen to this event todetect a user's network .

Important We strongly recommend reloading the page upon chain changes, unless you have a good reason not to:

provider // Or window.ethereum if you don't support EIP-6963. . on ("chainChanged" ,

(chainId)

=> window . location . reload ( ))

connect

interface

ConnectInfo

{ chainId :

string ; }

provider // Or window.ethereum if you don't support EIP-6963. . on ("connect" ,

handler :

(connectInfo : ConnectInfo)

=>

void) ; The provider emits this event when it's first able to submit RPC requests to a chain. We recommend listening to this event and using theisConnected() provider method to determine when the provider is connected.

disconnect

provider // Or window.ethereum if you don't support EIP-6963. . on ("disconnect" ,

handler :

(error : ProviderRpcError)

=>

void) ; The provider emits this event if it becomes unable to submit RPC requests to a chain. In general, this only happens due to network connectivity issues or some unforeseen error.

When the provider emits this event, it doesn't accept new requests until the connection to the chain is re-established, which requires reloading the page. You can also use theisConnected() provider method to determine if the provider is disconnected.

message

interface

ProviderMessage

{ type :

string ; data :

unknown ; }

provider // Or window.ethereum if you don't support EIP-6963. . on ("message" ,

handler :

(message : ProviderMessage)

=>

void) ; The provider emits this event when it receives a message that the user should be notified of. Thetype property identifies the kind of message.

RPC subscription updates are a common use case for this event. For example, if you create a subscription usingeth_subscribe , each subscription update is emitted as amessage event with atype ofeth_subscription .

Remove event listeners

removeListener

Use theremoveListener method to remove specific event listeners from anEventEmitter object. In the following exampleremoveListener is used to remove theconnect andaccountsChanged events:

// Use window.ethereum instead of provider if EIP-6963 is not supported.

// Add listeners provider . on ("_initialized" , updateWalletAndAccounts) provider . on ("connect" , updateWalletAndAccounts) provider . on ("accountsChanged" , updateWallet) provider . on ("chainChanged" , updateWalletAndAccounts) provider . on ("disconnect" , disconnectWallet)

// Remove individual listeners provider . removeListener ("connect" , updateWalletAndAccounts) provider . removeListener ("accountsChanged" , updateWallet) The first argument ofremoveListener is the event name, and the second argument is a reference to the function passed toon for the event.

removeAllListeners

You can useremoveAllListeners to remove all listeners from the event emitter at once. This method is helpful when you need to clean up all listeners simultaneously.

caution UseremoveAllListeners with caution. This method clears all event listeners associated with the emitter, not only the listeners set up by the application code. Using this method can unexpectedly clear important event handlers, interfere with scripts, and make debugging more complex. You can use theremoveListener method to safely remove specific listeners. // Use window.ethereum instead of provider if EIP-6963 is not supported.

// Add listeners provider . on ("_initialized" , updateWalletAndAccounts) provider . on ("connect" , updateWalletAndAccounts) provider . on ("accountsChanged" , updateWallet) provider . on ("chainChanged" , updateWalletAndAccounts) provider . on ("disconnect" , disconnectWallet)

// Remove all listeners provider . removeAllListeners ( ) In the provided code example,removeAllListeners is called to remove all event listeners attached to theprovider object. This cleanup function deletes any event listeners that are no longer needed.

Errors

All errors returned by the MetaMask provider follow this interface:

interface

ProviderRpcError

extends

Error

{ message :

string code :

number data ? :

unknown } Therequest() provider method throws errors eagerly. You can use the errorcode property to determine why the request failed. Common codes and their meaning include:

- 4001
-
    - The request is rejected by the user.
- -32602
-
    - The parameters are invalid.
- -32603
-
    - Internal error.

For the complete list of errors, seeEIP-1193 andEIP-1474 .

tip Theeth-rpc-errors package implements all RPC errors returned by the MetaMask provider, and can help you identify their meaning.