Access a user's accounts

User accounts are used in a variety of contexts in Ethereum, including as identifiers and forsigning transactions . To request a signature from a user or have a user approve a transaction, your dapp can access the user's accounts using theeth_requestAccounts RPC method.

note eth_requestAccounts internally callswallet_requestPermissions torequest permission to call the restrictedeth_accounts method. When accessing a user's accounts:

- Only
- initiate a connection request in response to direct user action, such as
- selecting aconnect button
- .
- Always
- disable the connect button while the connection request is pending.
- Never
- initiate a connection request on page load.

note You can also access users' accounts on somenon-EVM networks .

Create a connect button

Important This section describes how to create a single connect button. When connecting to multiple wallets, use theConnect to MetaMask guide to create multiple connect buttons. We recommend providing a button to allow users to connect MetaMask to your dapp. Selecting this button should calleth_requestAccounts to access the user's accounts.

For example, the following JavaScript code accesses the user's accounts when they select a connect button:

```
index.js // You should only attempt to request the user's account in response to user interaction, such as // selecting a button. Otherwise, you pop-up spam the user like it's 1999. If you fail to retrieve // the user's account, you should encourage the user to initiate the attempt. const ethereumButton =

document . querySelector ( ".enableEthereumButton" ) const showAccount =

document . querySelector ( ".showAccount" )

ethereumButton . addEventListener ( "click" ,

( )

=>

{ getAccount ( ) } )

// While awaiting the call to eth_requestAccounts, you should disable any buttons the user can // select to initiate the request. MetaMask rejects any additional requests while the first is still // pending. async

function

getAccount ( )

{ const accounts =

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

method :

"eth_requestAccounts"

} ) . catch ( ( err )

=>

{ if

( err . code

===

4001 )
```

{ // EIP-1193 userRejectedRequest error. // If this happens, the user rejected the connection request. console . log ("Please connect to MetaMask.") }

else

{ console . error (err) } }) const account = accounts [0] showAccount . innerHTML

= account } The following HTML code displays the button and the current account:

index.html

< button

class

" enableEthereumButton "

Enable Ethereum </ button

< h2

Account: < span

class

" showAccount "

</ span

</ h2

Handle accounts

Use theeth_accounts RPC method to handle user accounts. Listen to theaccountsChanged provider event to be notified when the user changes accounts.

The following code handles user accounts and detects when the user changes accounts:

index.js let currentAccount =

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

method :

"eth_accounts"

}) . then (handleAccountsChanged) . catch ((err)

=>

{ // Some unexpected error. // For backwards compatibility reasons, if no accounts are available, eth_accounts returns an // empty array. console . error (err) })

// Note that this event is emitted on page load. If the array of accounts is non-empty, you're // already connected. provider // Or window.ethereum if you don't support EIP-6963. . on ("accountsChanged" , handleAccountsChanged)

// eth_accounts always returns an array. function

handleAccountsChanged (accounts)

{ if

(accounts . length

===

0)

{ // MetaMask is locked or the user has not connected any accounts. console . log ("Please connect to MetaMask.") }

else

if

(accounts [0]

!== currentAccount)

{ // Reload your interface with accounts[0]. currentAccount = accounts [0] // Update the account displayed (see the HTML for the connect button) showAccount . innerHTML

= currentAccount } } note eth_accounts now returns the full list of accounts for which the user has permitted access to. Previously,eth_accounts returned at most one account in theaccounts array. The first account in the array will always be considered the user's "selected" account.

Disconnect a user's accounts

Sinceeth_requestAccounts internally callswallet_requestPermissions for permission to calleth_accounts , you can usewallet_revokePermissions to revoke this permission, revoking your dapp's access to the user's accounts.

This is useful as a method for users to log out (or disconnect) from your dapp. You can then usewallet_getPermissions to determine whether the user is connected or disconnected to your dapp.

Seehow to revoke permissions for an example.

Edit this page