Responses (2)

There are currently no responses for this story. Be the first to respond. You have 1 free member-only story left this

web3-react: Connect Users to MetaMask (or any wallet) From Your Frontend





Hello, in this post I'll be showing how to use a library called web3-react in order to connect users to MetaMask or any wallet from your frontend.



My plan is to show you the most simple implementation possible. It will

Connect to MetaMask

Connected with 0x95900aFDF214860740E85dEEF69B1b0E422b64eC



slightly better, but CSS is not important for this tutorial so I'll skip past that.

We'll want to add some text that says "Not connected" when that is the case $\,$ for the wallet and "Connected with <address>" when it is connected. For now just leave it as "Not connected" until we implement the logic in a bit. I'll show all this code below.

Now let's add the connector. Add a components folder. Inside add a wallet folder. Inside that add a file called Connectors.js

This is a good point to install the web3-react dependency that allows you to connect to MetaMask. So install it like so:

You can think of a connector as the thing connecting you to a certain wallet. So this one is for MetaMask, but there are others like the WalletConnect connector and the Portis connector.

Now import it in and simply export an injected connector object like so:

import (InjectedConnector) from '@web3-react/injected-con
export const injected = new InjectedConnector({
 supportedChainIds: [1, 3, 4, 5, 42],
}

I don't feel I can give a good enough definition personally, so If you want to know what chainIDs are, just google something like "crypto what are chain ids".

For this next file, I am going to give you the entire code and then explain each piece in the logical order that I coded it in. So, here is index.js:

```
port { useWeb3React } from "@web3-react/core
      import { injected } from "../components/wallet/Cor
      export default function Home() {
        const { active, account, library, connector, activate, deactivate } = useWeb3React()
       async function connect() {
          try {
  await activate(injected)
 10
         } catch (ex) {
 12
 13
 15
       async function disconnect() {
            deactivate()
 17
         } catch (ex) {
 19
            console.log(ex)
 20
 21
       }
 22
          <div className="flex flex-col items-center justify-center":</pre>
 24
           cbutton onClick={connect} className="py-2 mt-20 mb-4 text-lg font-bold text-white rou
{active ? <span>Connected with <b>{account}</b></span> : <span>Not connected</span>}
 26
 27
            <button onClick={disconnect} className="py-2 mt-20 mb-4 text-lg font-bold text-white</pre>
 28
          </div>
 29
4
                                                                                           F
index.js hosted with ♥ by GitHub
```

To understand where I figured this stuff out, it's all in the documentation.

Next, we need to set up the provider. Again, here's all the code and I'll explain it piece by piece:

Go to index.js again and use the useWeb3React hook to get all the needed values:

```
const { active, account, library, connector, activate, deactivate } = useWeb3React()
```

We will not use all of these values, but I thought I'd show them just so you know about them. Here's my explanations for each:

```
active: is a wallet actively connected right now?
account: the blockchain address that is connected
library: this is either web3 or ethers, depending what you passed in
connector: the current connector. So, when we connect it will be the
injected connector in this example
activate: the method to connect to a wallet
deactivate: the method to disconnect from a wallet
```

Inside the *connect* function, use the *activate* function with the injected connector as an argument. This is what initiates the connection to the user's MetaMask wallet:

```
async function connect() {
  try {
    await activate(injected)
  } catch (ex) {
    console.log(ex)
  }
}
```

However, at the moment, it would just connect but not show the user is connected visually. To do that, let's do some simple conditional templating. If the user's wallet is connected or active, then show their address. Otherwise, not connected:

```
{active ? <span>Connected with <b>{account}</b></span> : <span>Not connected</span>}
```

I'm also going to add a Disconnect button and function:

```
async function disconnect() {
  try {
    deactivate()
} catch (ex) {
    console.log(ex)
}
}
...
Cutton onClick=(disconnect) className="py-2 mt-20 mb-4 text-lq font-bold text-white
```

Users can now connect and disconnect their wallets to this application. After this, all sorts of crazy things can be done like sending tokens or interacting with smart contracts which is what opens up a world of possibilities.

If you have any questions, feel free to ask.

Follow my social media to keep up-to-date:

YouTube, Twitch, Twitter

Coinmonks

74 Q2 LJ

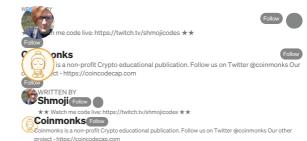
Sign up for Coinmonks

By Colmonols
A newsletter that brings you day's best crypto news, Technical analysis, Discount
Offers, and MEMEs directly in your inbox, by CoinCodeCap.com Take a look,
Medium sent you an
email at to complete
email at to complete evidencing in the complete evidence evidenc

By signing up, you will create a Medium account if you don't already have one. Review our <u>Privacy Policy</u> for more information about our privacy practices.

∰74 Q 2







More From Medium

React 17— Logicwind

GitHub" in a React App Princewill Iroka in Level Up Coding

"Login

to

ment = and **Functional Server** Componentswith in React Nginx and

Redis Jscrambler in HackerNoon.c

Animations API and React: The Fruits of My Labor

ntures in

Editor Asynchronous Starsatactions using Promises (the right way)

to

rm

Learn more.

Medium is an open platform where 170 million readers come to find insightful and dynamic thinking. Here, expert and undiscovered voices alike dive into the heart of any topic and bring new ideas to the surface. Learn

AboutWriteHelpLegalGet the Medium ap



Make Medium yours.

Follow the writers, publications, and topics that matter to you, and you'll see them on your homepage and in your inbox. Explore

Write a story on Medium.

If you have a story to tell, knowledge to share, or a perspective to offer — welcome home. It's easy and free to post your thinking on any topic. Start a blog