

There are currently no responses for this story. Be the first to respond.

You have 1 free member-only story left this month.

[Sign up for Medium and get an extra one](#)

Top highlight

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



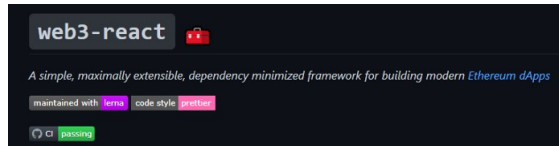
Shmoji

Follow

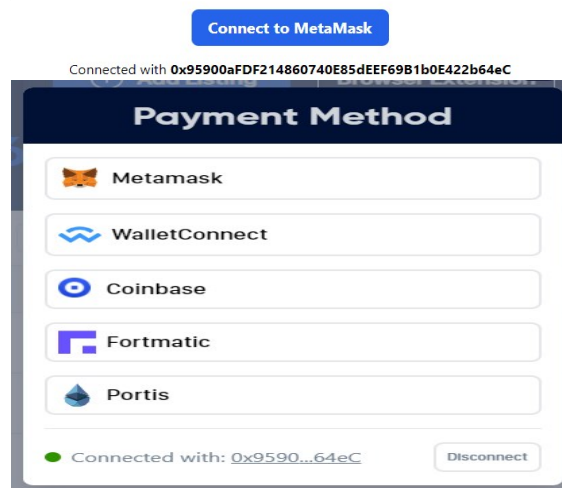
Aug 10 · 4 min read ★



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 look like this:



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:

```
yarn add @web3-react/injected-connector
```

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-connector'
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*:

```

1 import { useWeb3React } from "@web3-react/core"
2 import { injected } from "../components/wallet/Connectors"
3
4 export default function Home() {
5   const { active, account, library, connector, activate, deactivate } = useWeb3React()
6
7   async function connect() {
8     try {
9       await activate(injected)
10    } catch (ex) {
11      console.log(ex)
12    }
13  }
14
15  async function disconnect() {
16    try {
17      deactivate()
18    } catch (ex) {
19      console.log(ex)
20    }
21  }
22
23  return (
24    <div className="flex flex-col items-center justify-center">
25      <button onClick={connect} className="py-2 mt-20 mb-4 text-lg font-bold text-white rounded">
26        {active ? <span>Connected with <b>{account}</b></span> : <span>Not connected</span>}
27      <button onClick={disconnect} className="py-2 mt-20 mb-4 text-lg font-bold text-white rounded">
28        Disconnect
29    </div>
30  )
31 }

```

Index.js hosted with ❤ by GitHub [view raw](#)

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:

```

1 import './styles/globals.css'
2 import 'tailwindcss/tailwind.css'
3 import { Web3ReactProvider } from '@web3-react/core'
4 import Web3 from 'web3'
5
6 function getLibrary(provider) {
7   return new Web3(provider)
8 }
9
10 function MyApp({ Component, pageProps }) {
11   return (
12     <Web3ReactProvider getLibrary={getLibrary}>
13       <Component {...pageProps} />
14     </Web3ReactProvider>
15   )
16 }
17
18 export default MyApp

```

\_app.js hosted with ❤ by GitHub [view raw](#)

<Web3ReactProvider getLibrary={getLibrary}>  
 <Component {...pageProps} />  
 </Web3ReactProvider>

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)
  }
}
...
<button onClick={disconnect} className="py-2 mt-20 mb-4 text-lg font-bold text-white rounded">
  Disconnect

```

