Migration from RainbowKit

Rainbowkit is a popular wallet adapter library built on Wagmi. That said, it has several limitations in terms of flexibility, customizability etc. As your company scales, it makes sense to explore transitioning to Dynamic for wallet auth. In this guide, we'll cover how you can quickly swap out Rainbowkit for Dynamic.

A few notes

1. Rainbowkit uses wagmi for all functions. Dynamic can leverage wagmi as well, requiring a quick connector to coordinate between the libraries.
2. Rainbowkit requires that you manage your own authentication. Dynamic, on the other hand, comes with auth built in and will return a JWT if you select the connect and auth option.

Overview

To start, we will assume your current setup looks similar to this:

const App = ( ) => { return (< WagmiConfig client = { wagmiClient}

< RainbowKitProvider chains = { chains}

< YourRoutes />

< ConnectButton />

</ RainbowKitProvider

</ WagmiConfig

) ; } ; In this setup, you have 3 dependencies:

1. Ethers.js
2. Wagmi
3. Rainbowkit

Step 1: Set up the Dynamic SDK

Shell npm install @dynamic-labs/sdk-react-core Note: Dynamic comes with ethers.js as part of its package, meaning you can remove any independent ethers.js installation. You can read the full Dynamic set up guidehere .

Step 2: Set up the Dynamic Wagmi connector

Shell npm install @dynamic-labs/wagmi-connector You can read the full wagmi guidehere .

Step 3: Configure Dynamic instead of Rainbowkit

ReplaceRainbowKitProvider withDynamicContextProvider

Note: you'll need to grab a Dynamic environment ID from youdeveloper dashboard First, we will swap out the Rainbowkit provider with the Dynamic one.

const App = ( ) => { return (< WagmiConfig client = { wagmiClient}

< DynamicContextProvider settings = { { environmentId : "Enter your Environment ID here" , } }

< YourRoutes />

< ConnectButton />

</ DynamicContextProvider

</ WagmiConfig

) ; } ;

ReplaceWagmiConfig withDynamicWagmiConnector

As mentioned in the full wagmi guide above, theDynamicWagmiConnector replaceswagmiConfig in the Dynamic setup. Note that the Dynamic provider wraps the Wagmi connector, vs wagmi wrapping the Rainbowkit adapter.

const App = ( ) => { return (< DynamicContextProvider settings = { { environmentId : "Enter your Environment ID here" , } }

< DynamicWagmiConnector

< YourRoutes />

< ConnectButton />

</ DynamicWagmiConnector

</ DynamicContextProvider

) ; } ;

ReplaceConnectButton withDynamicWidget

Dynamic uses the Dynamic widget to manage wallets, connect, user profile etc. We will replace the Rainbowkit ConnectButton with it:

export const App = ( ) => { return (< DynamicContextProvider settings = { { environmentId : "Enter your Environment ID here" , } }

< DynamicWagmiConnector

< DynamicWidget />

< YourApp />

</ DynamicWagmiConnector

</ DynamicContextProvider

) ; } ; That’s it! you should now have Dynamic working instead of Rainbowkit.

Things to know

Authenticating with Dynamic vs Rainbowkit

As mentioned above, Dynamic comes built in with authentication. That means that as part of a Dynamic setup, you receive optional access to user management functions, as well as a JWT that contains verified credentials for the user.

Optionally, you can chose to keep the only connect-only experience of Rainbowkit. To do so, you can add to the settings `initialAuthenticationMode: ‘connect-only’. You can read more about connect-only vs authentication[here.]

Chains and customization

Rainbowkit configures chains within code using wagmi. Dynamic, on the other hand, pulls its configuration from your[developer dashboard] .

Was this page helpful?

Yes No [Send new users an XMTP message](#)[Migrating Users](#) [twitter](#) [linkedin](#) [slack](#) [Powered by Mintlify](#)