Migrating to Dynamic

Migration from one auth and embedded wallet provider to another can seem daunting. You have users, they might have wallets, and a subset of them will have assets in their wallets, making it a fun adventure of a migration. But have no fear! We can help. Below, we put together a guide on migration strategies you can think through depending on your use case.

When in doubt, talk to us. Migrations can get complex, and we'd love to help guide you through them. You can set up a call with us here We put together a simple flow chart to help you identify the right migration strategy:

Migration Path A: You only have a front-end "connect wallet" flow

Difficulty level: You can do this in your sleep

Common scenarios

You built your own front end wallet connector or use Rainbowkit/ConnectKit. There is no SIWE, no issuing JWTs, and mostly just a front end.

Recommended approach

You pretty much just swap out the modals. If you use wagmi, add our wagmi connector and follow our rainbowkit guide , but in general, no migration needed. After the initial update to the Dynamic modal, you can stay on our connect-only mode and not store any user info with Dynamic, or alternatively start adding more sophisticated features.

Migration Path B: You store wallet info somewhere and a wallet is a user

Difficulty level: You need at least one eye open

Common scenarios

You built your own front end wallet connector or use Rainbowkit/ConnectKit, and in addition implemented a SIWE on your own, authenticating users and creating sessions for them.

Recommended approach

For front end changes, you can follow path A above. In addition, you can use our create wallet endpoint to generate users and add wallet records to them. When they first log in, they will verify wallet ownership via SIWE (we provide that and return a JWT - you don't need to do this), and you're done.

Migration Path C: You have users, but no embedded wallets

Difficulty level: You need one cup of coffee

Common scenarios

You use Auth0, Firebase or another web2 user management system, or alternatively created something on your own. You store your users' emails and basic info.

Recommended approach

You'll need to import your users into Dynamic

1. Create a new user
2. , including their info capture/kyc information and their data
3. For any profile information that Dynamic does not yet explicitly support, you can add these in the user.metadata, which is a key-value object
4. When they log in, we verify their emails

And that's it. You're done. When your users log in next time around, we will verify their email addresses/social login information independently as part of the auth flow, and issue a JWT once you have successfully logged in.

Migration Path D: You have users, and they have embedded EOA wallets

Difficulty level: You need two cups of coffee

Common scenarios

You use an embedded wallets provider, or have built a simple KMS-based key management system on your own

Recommended approach

There are really two options here:

1. Keep current users in their existing embedded wallets and generate Dynamic wallets for new users
2. Migrate existing wallet users to Dynamic

Keep current users:

1. Switch to Dynamic auth

2. When user logs in with email, check if they have an existing account:

3. If so, redirect them to old login flow

4. If they don't, create a Dynamic embedded wallet for them

Migrate users:

1. You'll need to import your users into Dynamic from the previous section, and follow to migrate users using our API
2. When a user logs in, check if they have a wallet in the previous system
3. Check if wallet was ever used/is empty
4. If it isn't, prompt user to transfer funds to new Dynamic wallet (this can feel like an account upgrade flow for the users)
5. Next time that the user logs in, they go to their Dynamic wallet

Migration Path E: You have users, and they have AA wallets

Difficulty level: This is non-trivial

Considerations

1. Your users have an EOA embedded wallet, but it only serves as a signer for an AA layer (safe, zerodev etc).
2. Hence, you'll want to add the Dynamic embedded wallet as a second signer to the AA layer, or alternatively swap signers from the old embedded wallet one to the Dynamic one

Main activities You'll want to have the end user add their new Dynamic wallet as a second signer to the smart contract wallet, or alternatively have them replace the current signer with a second signer

Steps

This is non-trivial, and we recommend that we chat before starting this process.You can book time here

Was this page helpful?