

# Server-side verification

When an end user connects their wallet, you, the developer, get a [JSON Web Token \(JWT\)](#) that can be used to verify some claims about the end user, notably a proof of ownership over a wallet public address.

Upon authentication, we generate a JWT signed with a private key (using RS256 algorithm) that is unique to you. In turn, you can use the associated public key (found in the [API tab](#) of your developer dashboard) to ensure that the token is authentic and hasn't been tampered with. In other words, if a JWT issued by Dynamic can be successfully verified with your public key, the information it contains can be trusted.

You can do this in multiple ways.

## Option 1: Leverage [NextAuth](#)

If you are using Next.js, you can easily [integrate the NextAuth library with Dynamic](#) to perform server-side verification and then use a session client-side.

## Option 2: Leverage [Passport.js](#)

We offer an official [passport-dynamic](#) extension.

## Option 2: Do-It-Yourself Verification

1. Install the [node-jsonwebtoken](#)
2. package
3. Obtain your public key from Dynamic's API dashboard or through our [keys](#)
4. API endpoint.
5. Get the JWT through the Dynamic SDK with an [authToken](#)
6. .
7. Send the authToken to the server as a Bearer token

```
import { useEffect, useState } from "react"; export const useFetch = ( authToken : string | null ) => { const [ data, setData ] =
useState ( { } ) ; useEffect ( ( ) => { const fetchApi = async ( ) => { await fetch ( "http://localhost:9000/api" , { headers : {
Authorization : Bearer { authToken } , } , } ) . then ( response => response . json ( ) ) . then ( setData ) ; } fetchApi ( ) } , [
authToken ] ) ; return { data } ; } ; 1. Verify the JWT on your server with the node-jsonwebtoken 2. package
```

```
var jwt= require ( 'jsonwebtoken' ) ; jwt . verify ( token , publicKey , function ( err , decodedToken ) { console . log (
decodedToken ) } ) ;
```

Was this page helpful?

Yes No [Sign up with Branded Wallets](#) [Adding Captcha Protection](#) [twitter](#) [linkedin](#) [slack](#) [Powered by Mintlify](#)