Welcome to Authentication and Authorization in Node.js

Skills Network

After you are finished reading this document, you will be able to:

- Define authentication.
- Explain session-based, token-based, and passwordless authentication.
- Compare and contrast different types of authentications, including session-based, token-based, and passwordless.

The authentication process confirms a user's identity using credentials by validating who they claim to be. Authentication assure only those with valid credentials can access the system. Authentication is the responsibility of an application's backend.

Three popular authentication methods in Node.js include:

1. Session-based
2. Token-based
3. Passwordless

Let's explain a little bit about each of these methods and compare them.

Session-based

Session-based authentication is the oldest form of authentication technology. Typically, the flow of a session is as follows:

1. The user uses their credentials to log in.
2. The login credentials are verified against the credentials in a database. The database is responsible for storing which resou
3. The server creates a session with a session ID that is a unique encrypted string. The session ID is stored in the database.
4. The session ID is also stored in the browser as a cookie.
5. When the user logs out or a specified amount of time has passed, the session ID is destroyed on both the browser and the

Token-based

Token-based security entails two parts: authentication and authorization. Authentication is the process of providing credentials a credentials. Authorization refers to the process of using that token so the resource server knows which resources the user shou

Token-based Authentication

Token-based authentication uses access tokens to validate users. An access token is a small piece of code that contains inform and expirations that get passed from a server to the client. An ID token is an artifact that proves that the user has been authenti

The token contains three parts, the header, the payload, and the signature. The header contains information about the type of t payload contains user attributes, called claims, such as permissions, groups, and expirations. The signature verifies the token's during transit. A JSON web token, pronounced "jot" but spelled JWT, is an internet standard for creating encrypted payload data

A user's browser makes a call to an authentication server and gets access to a web application. The authentication server then client as an encrypted cookie. The ID token is then passed to the app on the web server as proof that the user has been auther

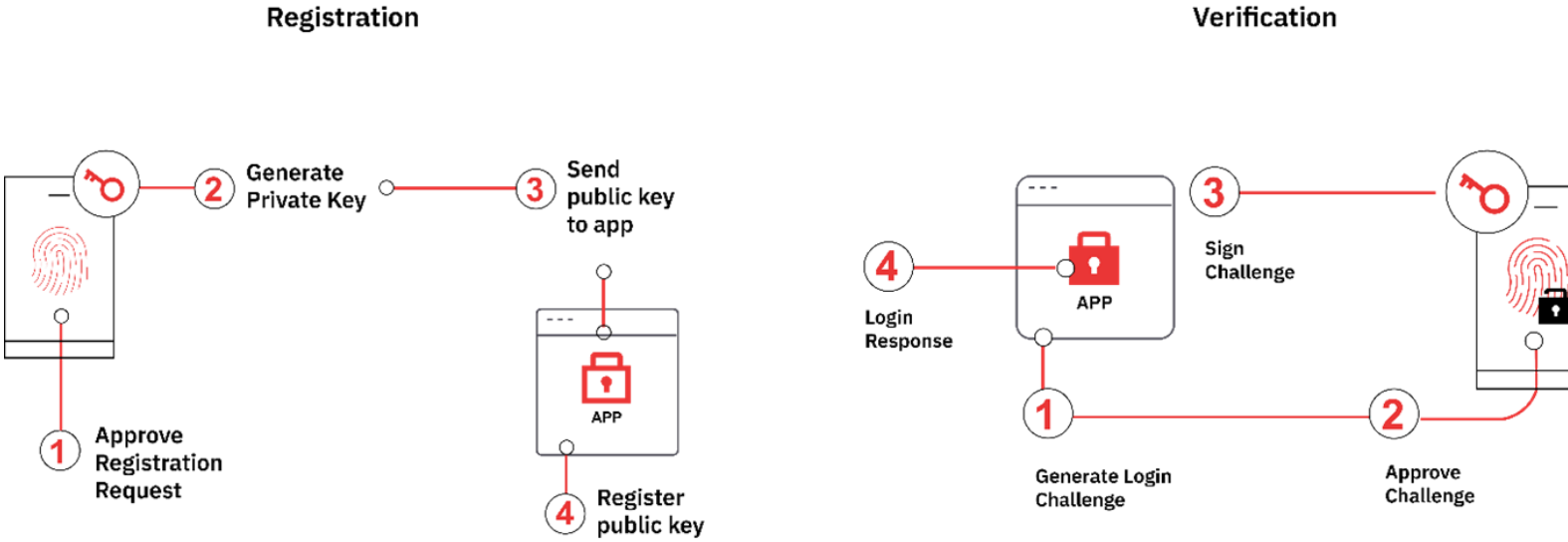
Token-based Authorization

This flowchart shows the workflow of a token through the authorization process.

Passwordless authentication is achieved using Public Key and Private Key Encryption. In this method, when a user registers fo
key/public key pair that utilizes a factor that proves their identity, as noted above.

The public key is used to encrypt messages, and the private key is used to decrypt them. The private key is stored on the user's
application and registered with a registration service.

Anyone may access the public key, but the private key is only known to the client. When the user signs into the application, the
requesting biometrics, sending a "magic link," or sending a special code via SMS, encrypting it with the public key. The private k
then verifies the sign-in challenge and accepts the response to authorize the user.



In this reading, you learned that:

- Authentication is the process of confirming a user's identity using credentials by validating who they claim to be.
- Session-based authentication uses credentials to create a session ID stored in a database and the client's browser. When
- Token-based authentication uses access tokens, often JWTs, that get passed between server and client with the data that i
- Passwordless authentication uses public/private key pairs to encrypt and decrypt data passed between client and server w