ACCESS and REFRESH TOKENS

The concept of access tokens and refresh tokens is fundamental in modern authentication systems, particularly in web development. Here's a breakdown of the process in a more formal manner:

- ***Access Tokens vs. Refresh Tokens***:
  Access tokens are short-lived tokens, typically valid for a short duration like one hour. On the other hand, refresh tokens are long-lived tokens, often valid for a longer period, such as 30 days.

- ***Token Issuance***:
  When a user successfully signs in to an application, the server generates both an access token and a refresh token. These tokens are then typically sent to the client, usually as HTTP cookies. Additionally, the refresh token is stored securely in the application's database, associated with the user's account.

- ***Token Usage***:
  Upon subsequent requests to the server, the client includes the access token in the request headers for authentication purposes. The server verifies the access token to ensure the user's identity and grants access to the requested resources if the token is valid.

- ***Token Refresh Mechanism***:
  If an access token expires, rather than forcing the user to log in again, the client can utilize the refresh token to obtain a new access token. This process typically involves making a request to a specific endpoint or API on the server dedicated to token refreshing. The server validates the refresh token and, if valid, issues a new access token to the client. Along with the new access token, a new refresh token may also be issued for security reasons. Both tokens are sent back to the client, often as HTTP cookies, and the refresh token in the database is updated to reflect the new token.

By implementing this token-based authentication flow, applications can achieve secure and efficient user authentication without requiring users to repeatedly log in within short timeframes.