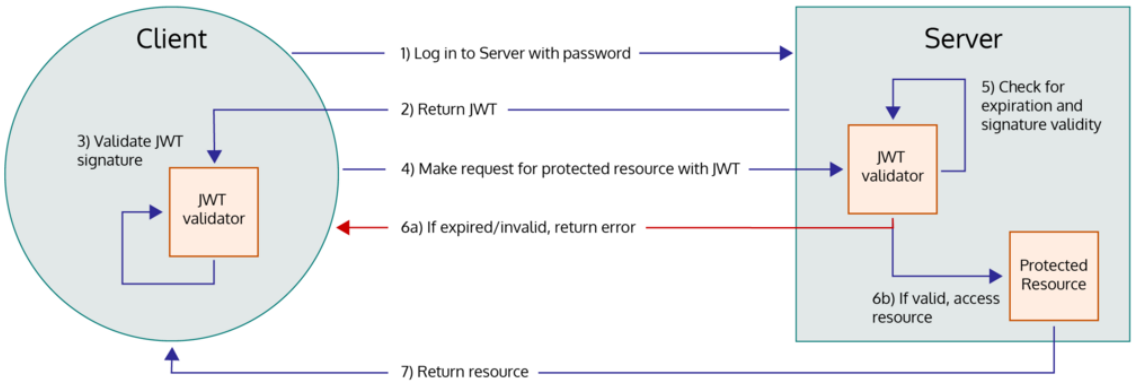
JSON Web Token (JWT)

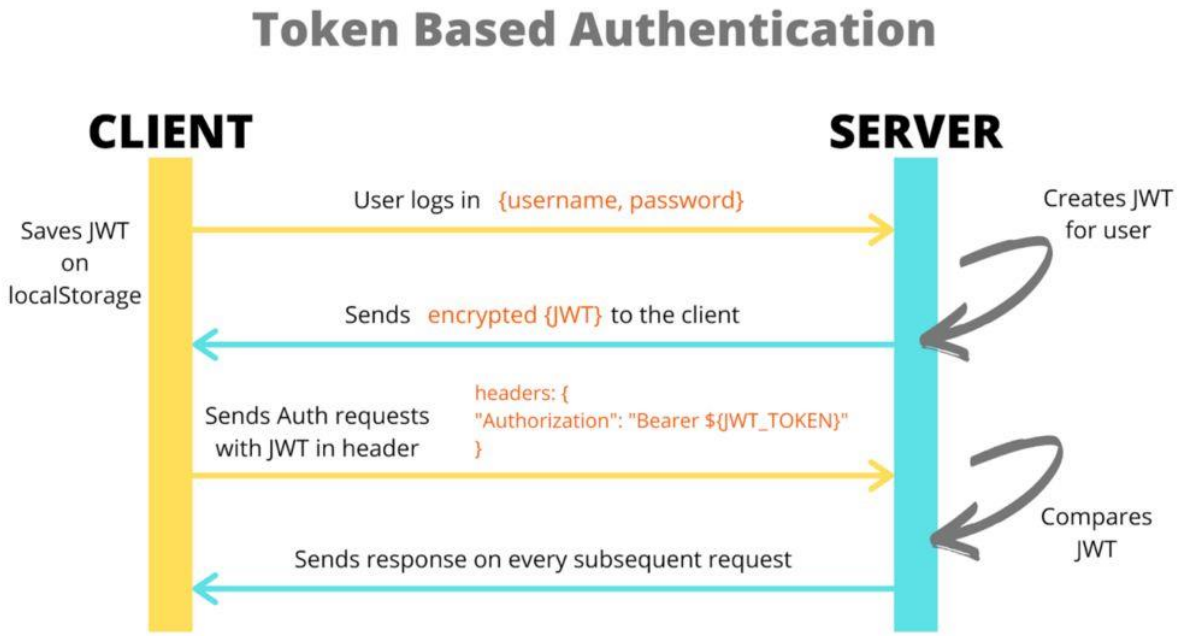
DEFINATION : JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

here is a diagram that shows how JWT works:

1.



2.



- The following is a brief explanation of each step:

1. The client requests authorization from the server.
2. The server generates a JWT token and sends it to the client.
3. The client sends the JWT token with all subsequent requests to the server.
4. The server validates the JWT token and checks the claims in the token.
5. If the JWT token is valid, the server grants access to the client.

The JWT token is a string that is made up of three parts:

1. Header : The header contains information about the token, such as the type of token and the algorithm that was used to sign the token.
2.Payload :The payload contains the claims, which are the data that is being transmitted between the client and the server.
3.Signature : The signature is used to verify the authenticity of the token.

The JWT token is signed using a secret key. This means that the token cannot be tampered with without knowing the secret key. This helps to ensure that the claims in the token are not changed.

JWT tokens are a popular way to implement authentication and authorization in web applications. They are a secure and efficient way to transmit data between the client and the server.

****

Below is an informative diagram illustrating the steps involved in a typical JWT-based

authorization process between a client and a server:

Explanation of each step:

1. The client initiates the authorization process by sending a request to the server.

2. The server generates a JWT (JSON Web Token) that contains relevant information, such as user identity and permissions.

3. The server sends the JWT token back to the client as a response to the authorization request.

4. The client includes the JWT token with all subsequent requests to the server, typically in the Authorization header.

5. Upon receiving a request, the server validates the JWT token to ensure its authenticity and integrity.

7. The server checks the claims within the JWT token, such as user roles or permissions, to determine if the client has the necessary access rights.

8. Based on the validation and claim checks, the server either grants or denies access to the requested resources.