JWT

JWT es un estándar abierto que define un método compacto y autónomo para transmitir información de forma segura entre partes, generalmente un cliente y un servidor, a través de un objeto JSON.

Un token JWT se compone de 3 partes separadas por puntos, ej:

- header.payload.signature

En primer lugar el header define el tipo de token y el algoritmo de cifrado:

- { "alg": "HS256", "typ": "JWT" }

Luego el payload (o carga útil) contiene los los datos “claims” que vendrían siendo características de algún usuario, ej:

- { "userId": 123, "role": "admin", "exp": 1694352000 }

Finalmente viene la parte de la firma o signature, el cual garantiza la integridad y autenticidad del código.

Mini ejemplo de código en un proyecto:

```
const jwt = require("jsonwebtoken");

// Generar token
const token = jwt.sign({ userId: 123, role: "admin" }, "secreto", { expiresIn: "1h" });

// Verificar token
jwt.verify(token, "secreto", (err, decoded) => {
  if (err) return console.log("Token inválido");
  console.log(decoded); // { userId: 123, role: "admin", iat:..., exp:... }
});
```

¿Cómo funciona?

1. **Login del usuario:**
    - El usuario se autentica (con email + contraseña).
    - El servidor valida y **genera un JWT** con los datos del usuario.

2. **Entrega del token:**
    - El cliente (navegador, app) guarda el token (normalmente en *localStorage* o *cookies seguras*).
3. **Acceso a recursos:**

En cada petición, el cliente envía el JWT en el **header**:

Authorization: Bearer <token>

    -
    - El servidor valida la firma del token y autoriza el acceso.
4. **Expiración:**
    - Los tokens suelen tener un tiempo de vida (exp) para mayor seguridad.
    - Cuando expira, el usuario debe pedir uno nuevo (refrescar sesión).