Certified Tech Developer
The Ultimate Degree

Entonces, ¿Qué es un JWT?

Bueno, en pocas palabras, es un estándar abierto de codificación, utilizado para transmitir información de manera segura entre dos partes. La manera en la que dicha información se transmite es a través de un objeto JSON (seguro recuerdas que vimos como el formato JSON se utilizaba para compartir información entre cliente y servidor), y su particularidad es que la información transmitida puede ser verificada ya que el JWT se encuentre firmado digitalmente.

Ahora bien, ¿cómo se compone un JWT?. Básicamente, la estructura más simple de un JWT consta de tres partes: Header, Payload y Signature

Ejemplo:

`xxxxx.yyyyy.zzzzz`

Veamos rápidamente cada una de dichas partes:

- Header: esta parte, contiene la información respecto del tipo de token (JWT), y el algoritmo de encriptación utilizado. Su estructura es la siguiente:

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

- Payload: es la parte más relevante desde el punto de vista de la autorización, ya que aquí se encontrará la información del usuario pudiendo incluir, por ejemplo, el rol que dicho usuario tiene dentro de la aplicación:

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

- Firma: es la parte que garantiza la autenticidad de la información incluida en el JWT, permitiendo su verificación.

Hasta aquí, vimos cómo se compone un JWT. Ahora, es momento de preguntarnos de qué manera podemos utilizar el mismo dentro del proceso de autorización que vimos anteriormente.

En líneas generales, cuando una persona inicia sesión en una aplicación determinada, el servidor verifica las credenciales ingresadas (nombre de usuario y contraseña). Si los mismos son correctos, el servidor **autentica** al usuario dentro de la aplicación, y envía un JWT como respuesta a la petición.

Dicho JWT, es entonces almacenado del lado del cliente, y enviado al servidor en cada nueva petición que se realice para acceder a un determinado servicio dentro de la aplicación. Ya que, como vimos más arriba, el token contiene la información del usuario (por ejemplo, su rol), el servidor puede acceder a dicha información al recibir la petición, y validar con ello si el

usuario se encuentra **autorizado** a realizarla. Si esto es así, el servidor procesará el pedido y enviará la respuesta correspondiente. Caso contrario, se devolverá un error indicado que la persona no se encuentra autorizada.

En resumen, JWT es una herramienta de gran utilidad para la comunicación entre cliente y servidor, ya que nos permite compartir información del usuario de manera segura y eficaz, y acceder a dicha información para validar los roles y permisos de cada persona que accede a nuestra aplicación.

Si quieres profundizar sobre este tema, te dejamos el link a la documentación oficial de JWT: https://jwt.io/#debugger-io