Curso CFP

CFP Programador full-stack

Introducción a Autenticación II

Agenda

- Repaso
- Mensajes Codificados
- JSON Web Token
 - Aspectos Básicos
 - Ejemplos
- Manejo de JWT
 - Lectura
 - Escritura
- Demo en Vivo
- Ejercicios

Repaso

Pasos para autenticarnos

- Hacer un POST pasando user/pass en el body
- Si el user/pass figura en la DB, devolver un token
- Hacer el resto de los request agregando el token como header
- La API debería verificar que el token exista y sea correcto para proceder a la funcionalidad

Repaso - Limitaciones

- Hasta el momento estamos devolviendo siempre el mismo token
- Podríamos personalizarlo, pero entraríamos a diseñar nuestro propio criterio
 - Tanto para armar como para escribir el token
 - Sobre todo en caso de agregar más cantidad de información
- Podemos ver el contenido del token a simple vista en el Postman
 - Cualquier persona malintencionada podría usar nuestro token

Mensajes Codificados (1)

- A veces no queremos que la información viaje de forma plana por un tema de seguridad
 - Por ejemplo si alguien nos mira el token que nos llega en el Postman, lo podría usar para acceder a un sistema con nuestro usuario
- Un mensaje codificado es aquél que está destinado para ciertas entidades
- Los destinatarios conocen la forma de decodificar el mensaje
 - De esta manera, un mensaje "ljkajsfasa" para nosotros no puede significar nada, pero para otra persona que sabe entenderlo, puede contener un mensaje

Mensajes Codificados (2)

- Otro ejemplo: el mensaje "ipmb dpnp bñebt"
- Es un mensaje que está codificado, una persona cualquiera no sabe qué quiere decir
- Ahora, sabemos que la forma de decodificarlo es retroceder un paso en el abecedario para cada palabra
 - "ipmb dpnp bñebt"
 - · "hola como andas"
- Este es un método básico para codificar y decodificar mensajes, existen otros más complejos

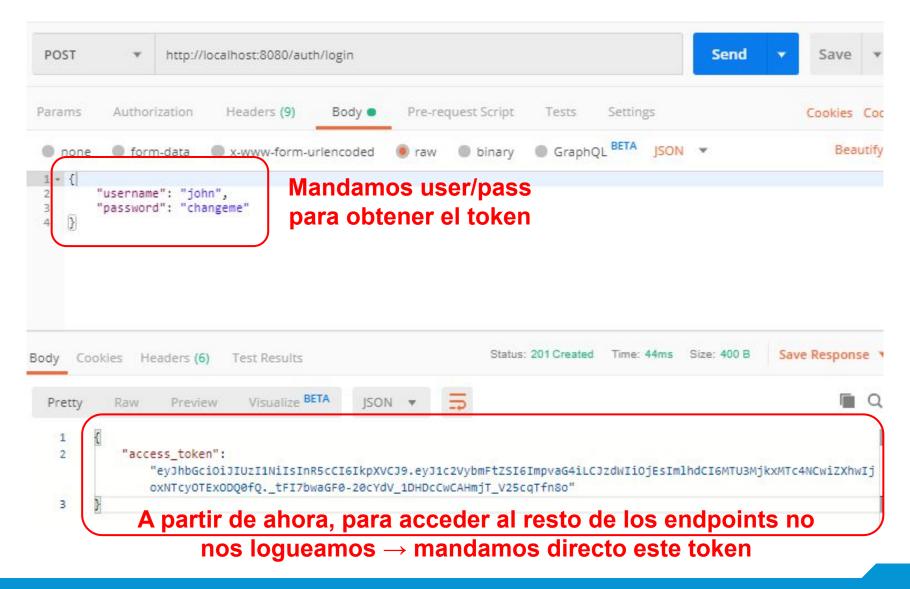
Mensajes Codificados (3)

- Existen varias forma de codificar y decodificar un mensaje: encriptación, firmas digitales, etc.
- Estos mecanismos normalmente funcionan mediante el proceso de hashing
- Una función de hash funciona como cualquier función
 - Toma una entrada y devuelve una salida
- Una función de hash recibe un mensaje y lo devuelve codificado o "hasheado"
 - Ejemplo: hola \rightarrow b221d9dbb083a7f33428d7c2a3c...
 - Se usó una función llamada SHA-256

JSON Web Token (JWT)

- Mecanismo muy utilizado para hacer manejo de usuarios (entre otras cosas)
- Consiste en un token en formato JSON
 - Podemos agregarle la información que queramos
- Está firmado → es decir que hay que decodificarlo para ver lo que dice
- Ejemplos de datos que contienen estos tokens
 - Identificador de usuario
 - Tipo de acceso que solicita
 - Expiración del token

Ejemplo de un JWT

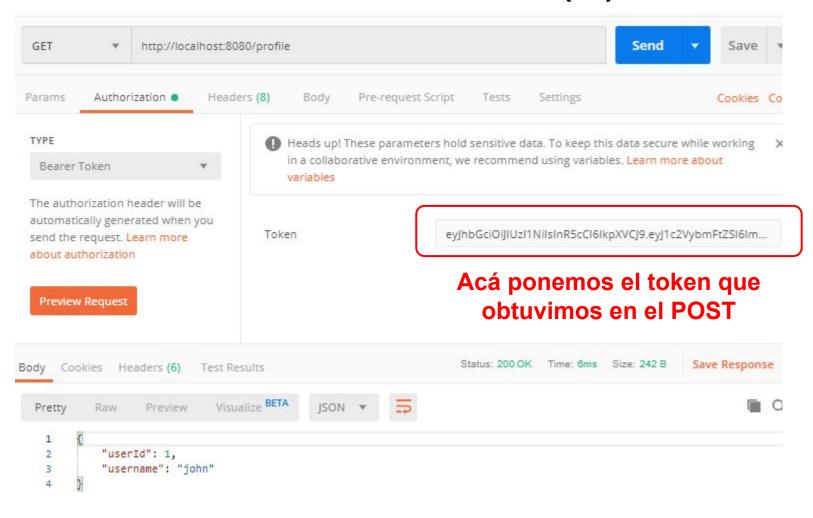


Manipulando un JWT

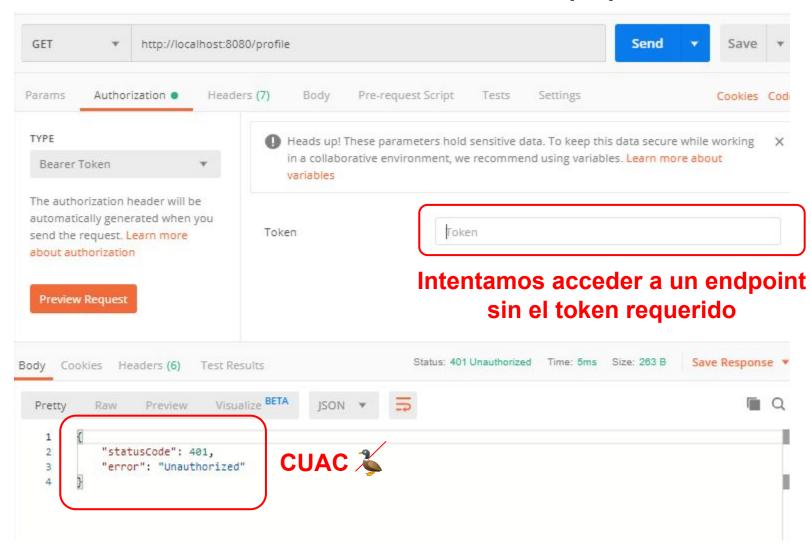
 Podemos ir a la página de JWT para ver qué es lo que tiene un token → https://jwt.io/#debugger

```
HEADER: ALGORITHM & TOKEN TYP
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
1c2VybmFtZSI6ImpvaG4iLCJzdWIi0jEsImlhdCI
                                                             "alg": "HS256",
6MTU3MjkxMTc4NCwiZXhwIjoxNTcyOTExODQ0fQ.
                                                             "tvp": "JWT"
tFI7bwaGF0-
20cYdV_1DHDcCwCAHmjT_V25cqTfn8o
                                                         PAYLOAD: DATA
                     En este caso figura el user en el
                                                             username": "john"
                   JWT → podemos poner más cosas
                                                             "iat": 1572911784.
                                                             "exp": 1572911844
                                                         VERIFY SIGNATURE
```

Usando el JWT (1)



Usando el JWT (2)



Demo en Vivo

- A partir de la API de prueba
 - Pegarle al POST para obtener un token
 - Acceder al endpoint GET con el token recibido
- Tener cuidado que la API de ejemplo genera tokens que expiran después de cierto tiempo
 - En ese caso, generarlo de vuelta
 - Los tokens siempre tienen que tener un tiempo de expiración por cuestiones de seguridad

Curso CFP

CFP Programador full-stack

Ejercicios

Ejercicios

- Partir de la API armada en la clase pasada, y agregar JWT en función al tutorial de NestJS https://docs.nestjs.com/techniques/authentication
- En la documentación se arma una API teniendo los usuarios hardcodeados en un arreglo interno
- La idea sería que esa información la tomemos desde la base de datos