

JWT: Librerías necesarias

Implementa la seguridad de tu API Rest con Spring Boot

Spring Security y JWT

- Spring Security, de forma *nativa*, nos permite utilizar JWT en el contexto del uso del framework OAuth2.0
- Si queremos implementar la autenticación basada en JWT con el mecanismo que hemos visto en la lección anterior, necesitamos alguna librería externa.

JJWT: Java JWT

- Java / Android
- Integrable vía Maven/Gradle
- Uso ampliamente extendido
- Actualizada recientemente y con frecuencia

JJWT: funcionalidades

- Construir un token con sus diferentes partes:

String jws =

Jwts.builder().setSubject("Joe").signWith(key).compact();

- Construimos un JWT con un *claim sub* con el valor Joe
- Firmamos el JWT con un algoritmo adecuado (*HMAC-SHA256*)
- Lo compactamos en un *String*.

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWQ1IiwiaWF0IjoxNjQ0MjQ0MDA0LCJ1b250aWQiOiJkb2UifQ.1KP0SsvENi7Uz1oQc07aXTL7kpQG5jBNlybqr60AID4

JJWT: funcionalidades

- Parsear un token, verificando así si está correctamente firmado
- Comprobar si
 - Ha expirado (*ExpiredJwtException*)
 - No está bien formado (*MalformedJwtException*)
 - Si la firma no es válida (*SignatureException*)