JWT Authentication

1. JSON web token
2. Used to Authenticate users
3. it gain popularity because of Token, no any cookies and no any session
4. when request first times come to Server, then server validate the token and if the validation request fails then request is rejected.

Steps:

1. check the JWT is well formed or not
2. check signature
3. validate the standard claims
4. Check the client permission

Structure:

X.Y.Z

where,

X = Algo + token Type also known as header

Y = Payload(data)

Z = Verified Signature

JWT Authentication Flow:

Client	Server

1. POST / authentication with username and password

——————————————————————————————————→

2. Validate the username

and password Generate

the JWT using secret key

3. Return the generated JWT

←

---

4. GET/ data with JWT in the header

→

5. Validate JWT using

secret key

6. Return the Response

←

Dependency Added:

jjwt-api, jjwt-impl, jjwt-jackson

Steps of Implementation:

1. make sure spring-boot-starter-security is there in pom.xml
2. Create class JWTAthenticationEntryPoint that implements AthenticationEntryPoint. Method of this class is called whenever as exception is thrown due to unauthenticated user try to access the source that requires authentication.
3. create JWTHelper class this contains method related to perform operation with JWT token like generatedToken, validateToken etc.

- Claims: A data store inside JWT containing information like username, expiration date, etc.
- secret_key: Used to verify and sign the JWT.
- <T> → Generic return type. Function<Claims, T> → Takes Claims and returns specific data (username, expiration, role).tionm, role)

Decode JWT Safely:

- `Jwts.parser()` → creates JWT parser
- `setSigningKey(secret)` → verifies token signature

- `parseSignedClaims(token)` → parses JWT
- `getPayload()` → returns claims data

4. Create JWTAuthenticationFilter that extends OncePerRequestFilter and override method and write the logic to check the token is comming is header. we have 5 simple steps to import logic.

- Get token form request
- Validate token
- GetUsername form token
- Load user Association with this token
- set authentication

5. Configure Spring Security is Configure file.