Date : 15-Jun-21
Spring Boot 9AM
Mr. RAGHU

----

*) CSRF Attack :-
CSRF - Cross Site Request Forgery

*) A Request object/link is create by hacker and executed by enduser,
server will process request by thinking that end user made it!!.

=> Spam Emails, ads in unknows sites, unknown URLs using whatsapp and messages
..etc

*) Solution :- Spring Security has provided one unique ID for every request
processing, that ID is specific to user/client(browser).

Attacker/Hacker may create request, but he cant add id, bcoz id are specific
to user, he may send his request created but not his ID.

----

*) Every Request is given to 'CsrfFilter' (OncePerRequestFilter)
=> it is creating one new Token and saving it using 'CsrfTokenRepository'
for the time first time, same token is converted into one hidden input
using class 'CsrfTokenHiddenInputFunction' as
ex: <input type="hidden" name="_csrf" value="3903ca06-0f9a-4bb0-85e2-1906a1074c9e"/>
and sent to browser as response attribute

*) 2nd request onwards 'CsrfFilter' is trying to check token exist or not?
by using 'CsrfTokenArgumentResolver'
If token not exist, again creat new token and send to /login
using 'CsrfToken'.

*) if token exist then 'CsrfRequestDataValueProcessor' validates it using
CsrfTokenRepository#load method, if valid continue same request
else goto /login (with new token)

Q) How can we disable these CSRF Token?
A)

```
   WebSecurityConfigurerAdapter#configure(HttpSecurity http){
              http
                .csrf().disable()
                .authorizeRequests()
                .antMatchers(..)
                ...
   }
----------------------------------------------------------------------
*) JWT :  JSON Web Token | Stateless Token based services


<properties>
                <maven.compiler.source>1.8</maven.compiler.source>
                <maven.compiler.target>1.8</maven.compiler.target>
        </properties>
        <dependencies>
                <dependency>
                        <groupId>io.jsonwebtoken</groupId>
                        <artifactId>jjwt</artifactId>
                        <version>0.9.1</version>
                </dependency>

                <dependency>
                        <groupId>javax.xml.bind</groupId>
                        <artifactId>jaxb-api</artifactId>
                        <version>2.3.0</version>
                </dependency>


        </dependencies>
----------------------------------------------------------------------
--
package in.nit;

import java.util.Date;
import java.util.concurrent.TimeUnit;

import io.jsonwebtoken.Claims;
import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;

public class Test {

        public static void main(String[] args) {
                String secret ="nitraghu";

                String token = Jwts.builder()
                                .setId("23355AA")
                                .setSubject("sam")
                                .setIssuer("NIT")
                                .setIssuedAt(new Date())
                                .setExpiration(new
Date(System.currentTimeMillis() + TimeUnit.MINUTES.toMillis(15)))
                                .signWith(SignatureAlgorithm.HS512,
secret)
                                .compact();

                System.out.println(token);

                Claims c = Jwts.parser()
                                .setSigningKey(secret)
                                .parseClaimsJws(token)
```

```
                                        .getBody()       ;

                    System.out.println(c.getId());
                    System.out.println(c.getSubject());
                    System.out.println(c.getIssuer());
          }
}
```