**SPRING SECURITY**

**When a user logs into something, the user receives a token. Every time the user makes a request, that token is appended to the request and thus the server will know who issued that request.**

**I can control when the token expires. I can control how its constructed.**

**That said, we need some kind of interceptor / filter in the backend, through which every request will pass. The modus operandi of the filter is something like : I get the request -> I look at the token -> If the token is valid, I let the user do whatever it wants with the request.**

**That filter will be activated before any entry point of our app.( So before the controller). We also have an exception here. For the SIGN IN or SIGN UP requests, that filter shouldn’t be activated, since there is not yet any token for that user.**

**We can start by defining a class with our paths:**

**Text

Description automatically generated**

**Dependencies**

<dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt</artifactId>  
 <version>0.9.1</version>  
</dependency>

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-security</artifactId>  
 <version>2.3.9.RELEASE</version>  
</dependency>

<dependency>  
 <groupId>javax.validation</groupId>  
 <artifactId>validation-api</artifactId>  
 <version>2.0.0.Final</version>  
</dependency>

**(WebSecurityConfig)**

**We start by creating a new package ( security) and creating inside of it a new class (WebSecurityConfig)**

**App.properties**

# SPRING SECURITY 11 ( cu ce encodam si in cat timp sa expire, 24H in cazul nostru)  
app.jwtSecret=AWs$1@234wa  
app.jwtExpirationMs=86400000

**Sau le declaram in jwtUtils, luand valorile deja declarate in App.properties**

**Text

Description automatically generated**

**We will use DTOs for the transferring of data.**

**DTOs are of 2 types:**

1. **Response DTOs**
2. **Request DTOs**