**Spring Boot - JWT**

JwtAuthFilter -> The first thing that gets executed when an HTTP request occurs - want to be active everytime the server gets a request

* filterChain: the chain of responsibility design pattern (contains the filters within the chain that should be executed)

UserDetailsService -> tries to fetch information from the database (as a claim that will be extracted from the token)

Validate JWT -> if we get the user from the database, a validation process will start

* If the token is valid, the SecurityContextHolder will be updated
* Otherwise a HTTP 403 will be send to the client

UserDetails -> (uses this object when spring security starts) an interface that contains a bunch of methods - implements the Class Employee in order to make it from the start a spring object

Bean -> In Spring, the objects that form the backbone of your application and that are managed by the Spring IoC container are called beans. A bean is an object that is instantiated, assembled, and otherwise managed by a Spring IoC container.

* Simply put, Inversion of Control (IoC) is a process in which an object defines its dependencies without creating them. This object delegates the job of constructing such dependencies to an IoC container.

Claims -> encoded as json objects that are signed using json signature (e.g. email, username etc) - contained in the payload