

 11

O que aprendemos?

Nesta aula, aprendemos que:

- Para utilizar o módulo do Spring Security, devemos adicioná-lo como dependência do projeto no arquivo **pom.xml**;
- Para habilitar e configurar o controle de autenticação e autorização do projeto, devemos criar uma classe e anotá-la com `@Configuration` e `@EnableWebSecurity` ;
- Para liberar acesso a algum *endpoint* da nossa API, devemos chamar o método `http.authorizeRequests().antMatchers().permitAll()` dentro do método `configure(HttpSecurity http)` , que está na classe `SecurityConfigurations` ;
- O método `anyRequest().authenticated()` indica ao Spring Security para bloquear todos os *endpoints* que não foram liberados anteriormente com o método `permitAll()` ;
- Para implementar o controle de autenticação na API, devemos implementar a interface `UserDetails` na classe `Usuario` e também implementar a interface `GrantedAuthority` na classe `Perfil`;
- Para o Spring Security gerar automaticamente um formulário de login, devemos chamar o método `and().formLogin()` , dentro do método `configure(HttpSecurity http)` , que está na classe `SecurityConfigurations` ;
- A lógica de autenticação, que consulta o usuário no banco de dados, deve implementar a interface `UserDetailsService` ;
- Devemos indicar ao Spring Security qual o algoritmo de `hashing` de senha que utilizaremos na API, chamando o método `passwordEncoder()` , dentro do método `configure(AuthenticationManagerBuilder auth)` , que está na classe `SecurityConfigurations` .