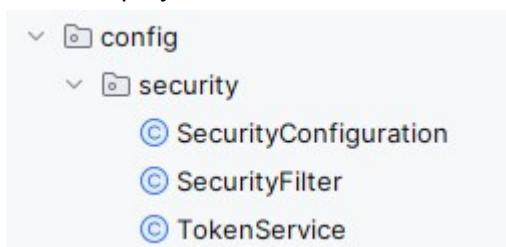


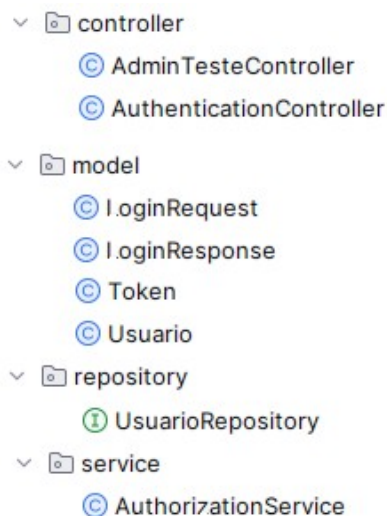
1. Incluir no “pom.xml” as dependências abaixo:

```
<!-- ===== -->
<!-- Dependencies of SECURITY -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
  <groupId>com.auth0</groupId>
  <artifactId>java-jwt</artifactId>
  <version>4.4.0</version>
</dependency>
<!-- ===== F ===== -->
```

2. Incluir no projeto de vocês as classes:



3. Incluir no projeto as classes:



4. Criar no banco de dados a seguinte tabela:

```
CREATE TABLE `usuario` (
  `id` int NOT NULL AUTO_INCREMENT,
  `nome` varchar(70) NOT NULL,
  `email` varchar(100) NOT NULL,
  `senha` varchar(250) NOT NULL,
  `role` varchar(45) NOT NULL,
  `dtAlteracao` datetime DEFAULT NULL,
  `dtCriacao` datetime NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `email_UNIQUE` (`email`)
)
```

5. Configurar quais são as rotas e como serão suas permissões. Fazer isto na classe “SecurityConfiguration”

Rotas:

```
public class SecurityConfiguration{
    private final SecurityFilter securityFilter;

    public static final String AUTH_URI = "/api/auth/"; 1 usage
    public static final String ADMIN_URI = "/api/v1/admin/"; 1 usage
    //public static final String PUBLIC_URI = "/api/v1/public/";
```

Permissões:

```
.authorizeHttpRequests(
    AuthorizationManagerRequestMat... authorize->authorize
        .requestMatchers(AUTH_URI + "**").permitAll()
        .requestMatchers(ADMIN_URI + "**").hasRole("ADMIN")
        .anyRequest().permitAll()
)
```

6. Teste o acesso à classe “AdminTesteController” com e sem o TOKEN.