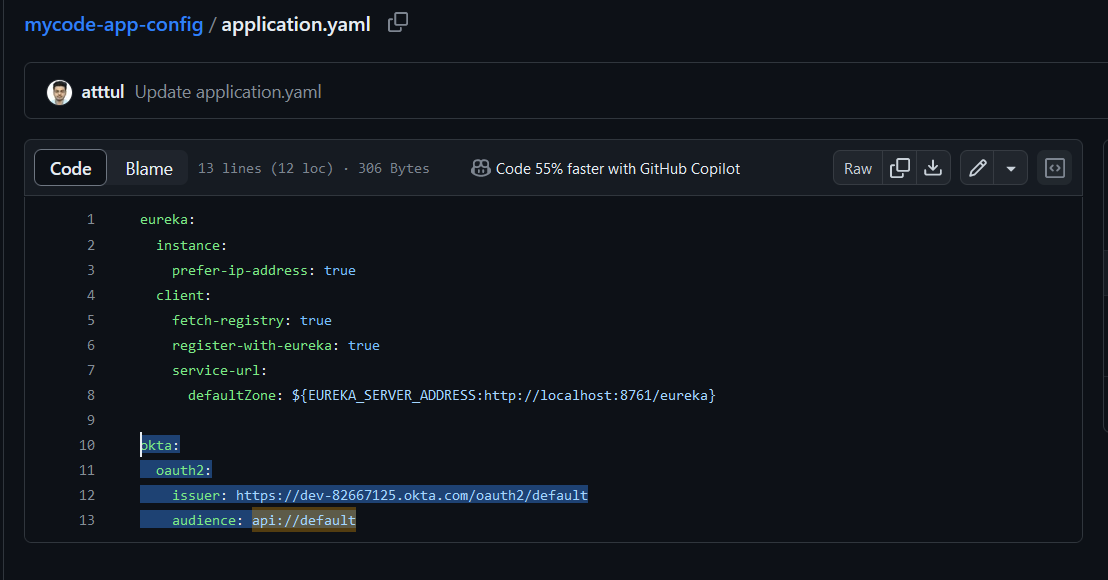
1. <https://start.spring.io> and add 2 dependencies in API Gateway:
   * 1. Spring Security
     2. OKTA
     3. OAuth2 Client
2. Open GitHub and add these lines:

okta:

oauth2:

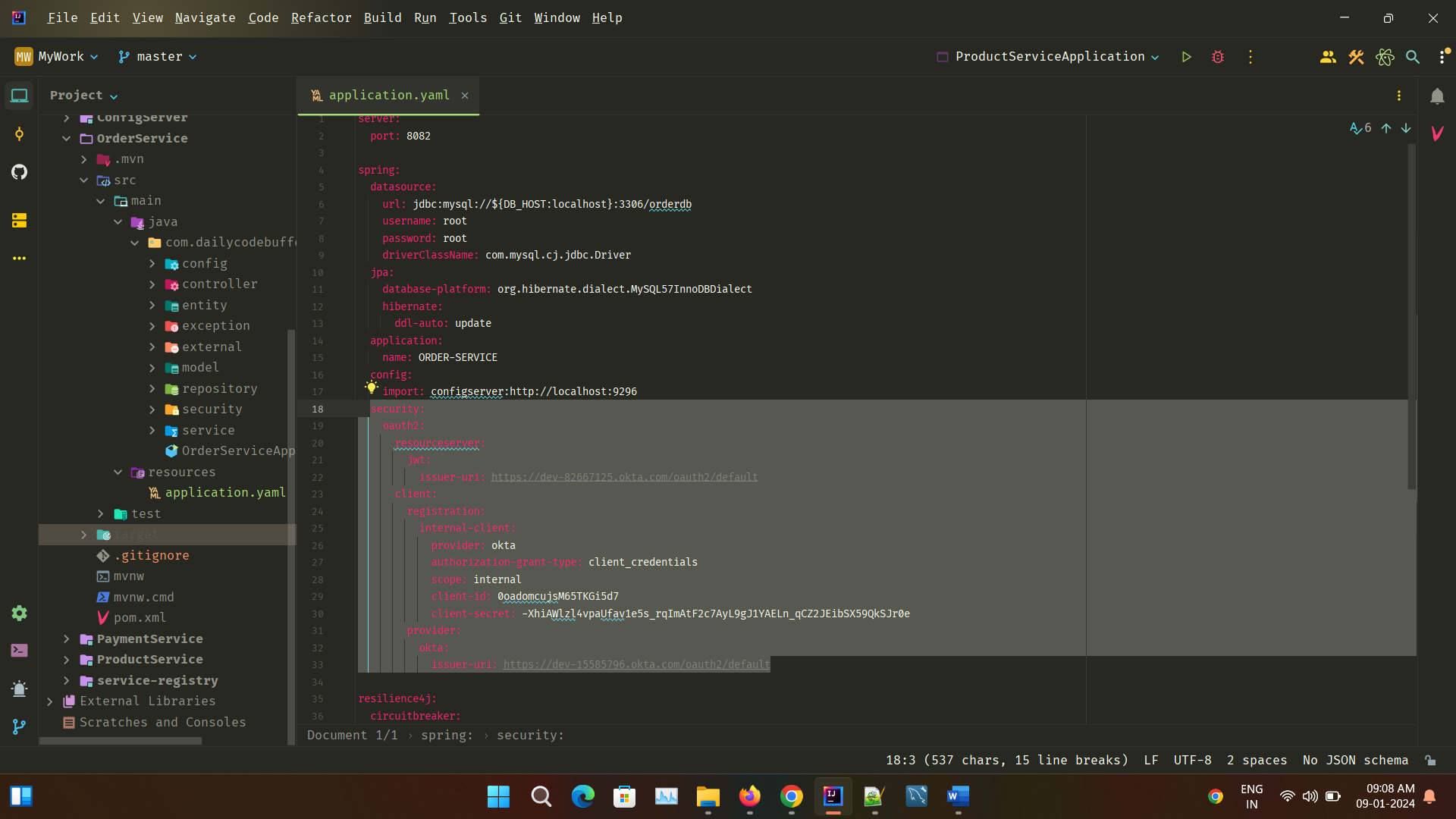
issuer: https://dev-82667125.okta.com/oauth2/default

audience: api://default



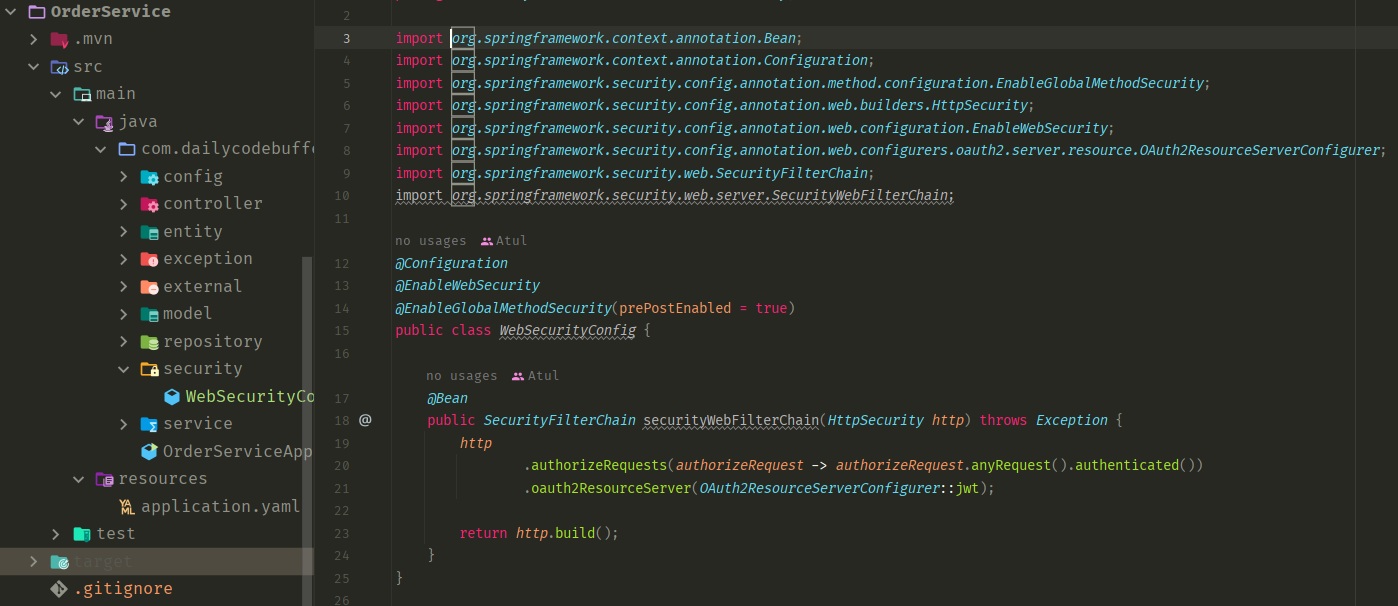
1. Application.yaml file: (Here we need to give the information of client registration, This is going to be registered as a Client, which will call other services)

Spring:  
security:  
 oauth2:  
 resourceserver:  
 jwt:  
 issuer-uri: https://dev-82667125.okta.com/oauth2/default  
 client:  
 registration:  
 internal-client:  
 provider: okta  
 authorization-grant-type: client\_credentials  
 scope: internal  
 client-id: 0oadomcujsM65TKGi5d7  
 client-secret: -XhiAWlzl4vpaUfav1e5s\_rqImAtF2c7AyL9gJ1YAELn\_qCZ2JEibSX59QkSJr0e  
 provider:  
 okta:  
 issuer-uri: https://dev-15585796.okta.com/oauth2/default



1. Create a “security” package and a class “WebSecurityConfig.java”:

*@Configuration  
@EnableWebSecurity  
@EnableGlobalMethodSecurity*(prePostEnabled = true)  
public class *WebSecurityConfig* {  
  
 *@Bean* public *SecurityFilterChain* securityWebFilterChain(*HttpSecurity http*) throws *Exception* {  
 *http* .authorizeRequests(*authorizeRequest* -> *authorizeRequest*.anyRequest().authenticated())  
 .oauth2ResourceServer(*OAuth2ResourceServerConfigurer*::jwt);  
  
 return *http*.build();  
 }  
}



1. Open OrderController.java and add @PreAuthorize(“has Authority(‘Admin’))

