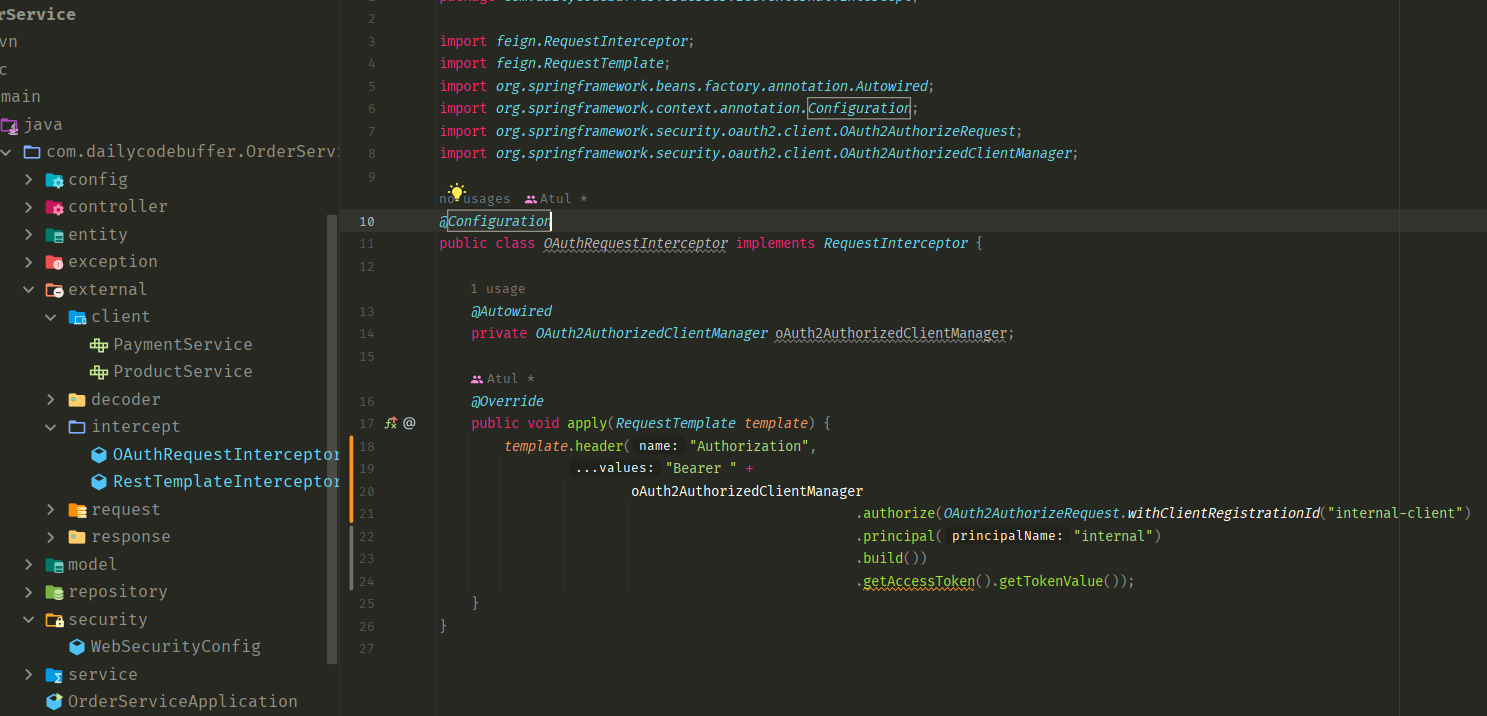
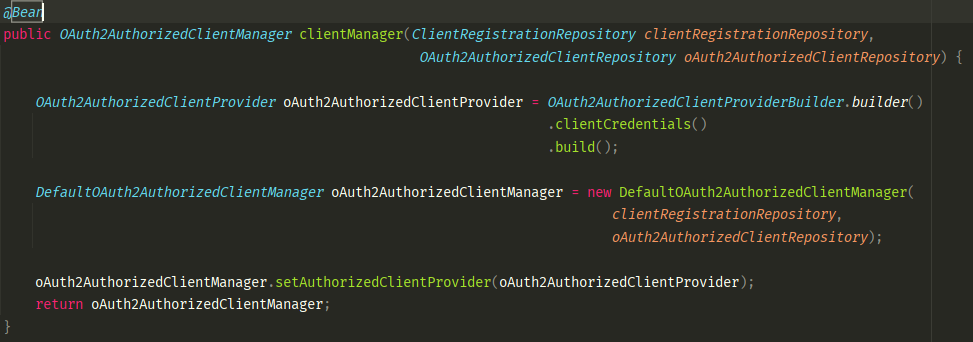
1. Create a package “external.intercept”
2. Create a class OAuthRequestInterceptor.java (this class is for Feign Client)

*@Configuration*public class *OAuthRequestInterceptor* implements *RequestInterceptor* {  
  
 *@Autowired* private *OAuth2AuthorizedClientManager* oAuth2AuthorizedClientManager;  
  
 *@Override* public void apply(*RequestTemplate template*) {  
 *template*.header("Authorization",  
 "Bearer " +  
 oAuth2AuthorizedClientManager  
 .authorize(*OAuth2AuthorizeRequest*.*withClientRegistrationId*("internal-client")  
 .principal("internal")  
 .build())  
 .getAccessToken().getTokenValue());  
 }  
}



1. Open main application file for one bean creation.

*@Bean*public *OAuth2AuthorizedClientManager* clientManager(  
 *ClientRegistrationRepository clientRegistrationRepository*,  
 *OAuth2AuthorizedClientRepository oAuth2AuthorizedClientRepository*) {  
 *OAuth2AuthorizedClientProvider* oAuth2AuthorizedClientProvider  
 = *OAuth2AuthorizedClientProviderBuilder* .*builder*()  
 .clientCredentials()  
 .build();  
  
 *DefaultOAuth2AuthorizedClientManager* oAuth2AuthorizedClientManager  
 = new DefaultOAuth2AuthorizedClientManager(  
 *clientRegistrationRepository*,  
 *oAuth2AuthorizedClientRepository*);  
  
 oAuth2AuthorizedClientManager.setAuthorizedClientProvider(  
 oAuth2AuthorizedClientProvider  
 );  
  
 return oAuth2AuthorizedClientManager;  
}



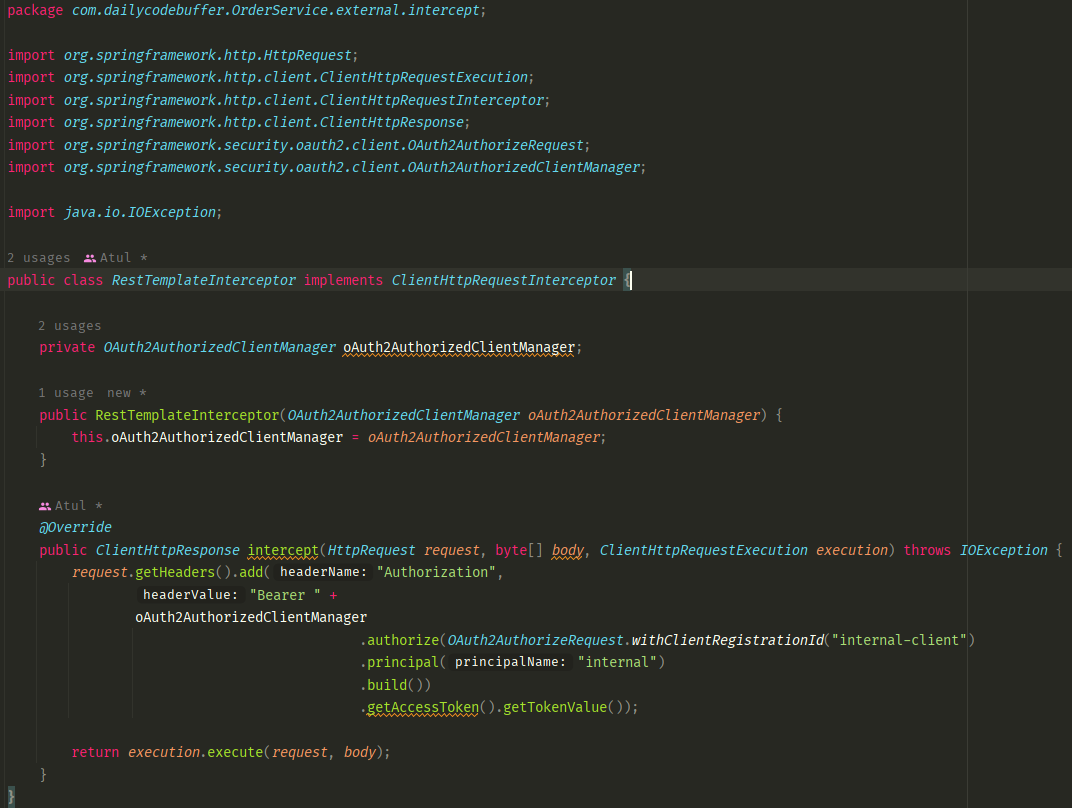
(We can only use feign client interceptor, no need of restTemplate interceptor, if we didn’t use it.)

Now, we have added the interceptor.

Now whatever the request we are going to pass from OrderService, it will send those request with those particular headers to the relevant services(Product or Payment)

1. For RestTemplate: Create a class: OAuth2AuthorizedClientManager.java

public class *RestTemplateInterceptor* implements *ClientHttpRequestInterceptor* {  
  
 private *OAuth2AuthorizedClientManager* oAuth2AuthorizedClientManager;  
  
 public RestTemplateInterceptor(*OAuth2AuthorizedClientManager oAuth2AuthorizedClientManager*) {  
 this.oAuth2AuthorizedClientManager = *oAuth2AuthorizedClientManager*;  
 }  
  
 *@Override* public *ClientHttpResponse* intercept(*HttpRequest request*, byte[] *body*, *ClientHttpRequestExecution execution*) throws *IOException* {  
 *request*.getHeaders().add("Authorization",  
 "Bearer " +  
 oAuth2AuthorizedClientManager  
 .authorize(*OAuth2AuthorizeRequest*.*withClientRegistrationId*("internal-client")  
 .principal("internal")  
 .build())  
 .getAccessToken().getTokenValue());  
  
 return *execution*.execute(*request*, *body*);  
 }  
}



1. Then create a bean for it in main application class.

*@Autowired*private *ClientRegistrationRepository* clientRegistrationRepository;  
*@Autowired*private *OAuth2AuthorizedClientRepository* oAuth2AuthorizedClientRepository;  
*@Bean  
@LoadBalanced*public *RestTemplate* restTemplate() {  
 *RestTemplate* restTemplate = new RestTemplate();  
 restTemplate.setInterceptors(*Arrays*.*asList*(new RestTemplateInterceptor(clientManager(  
 clientRegistrationRepository,  
 oAuth2AuthorizedClientRepository))));  
  
 return restTemplate;  
}

