**Primer Parcial – Especialización en Backend II**

**Alumna: Ornella Quarracino**

**Junio 2023**

Contenido

[Configuración Keycloak 2](#_Toc137218125)

[Reino: PrimerParcialOQ 2](#_Toc137218126)

[Roles de Reino 2](#_Toc137218127)

[Clientes 2](#_Toc137218128)

[Users 2](#_Toc137218129)

[Configuración del Gateway 3](#_Toc137218130)

[Configuración del ms-bill 4](#_Toc137218131)

[Configuración de Eureka 5](#_Toc137218132)

[Ingreso mediante navegador 5](#_Toc137218133)

[Ingreso mediante Postman 7](#_Toc137218134)

[Versiones utilizadas 9](#_Toc137218135)

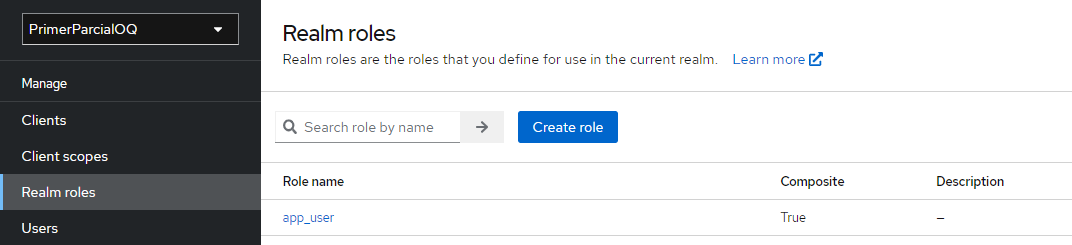
# Configuración Keycloak

Se levantó en el puerto 8080

## Reino: PrimerParcialOQ

Se eligió dicho nombre para que sea representativo al trabajo, considerando que se trata de la primera instancia de evaluación junto a mis iniciales.

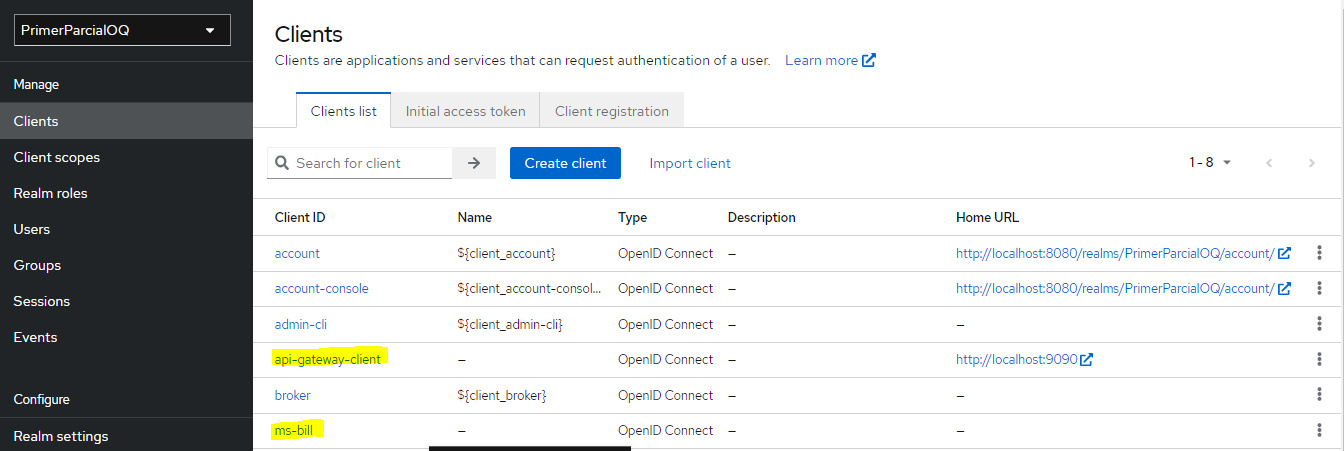
## Roles de Reino



Se generó el rol de reino llamado app\_user.

## Clientes

Se generaron dos clientes:

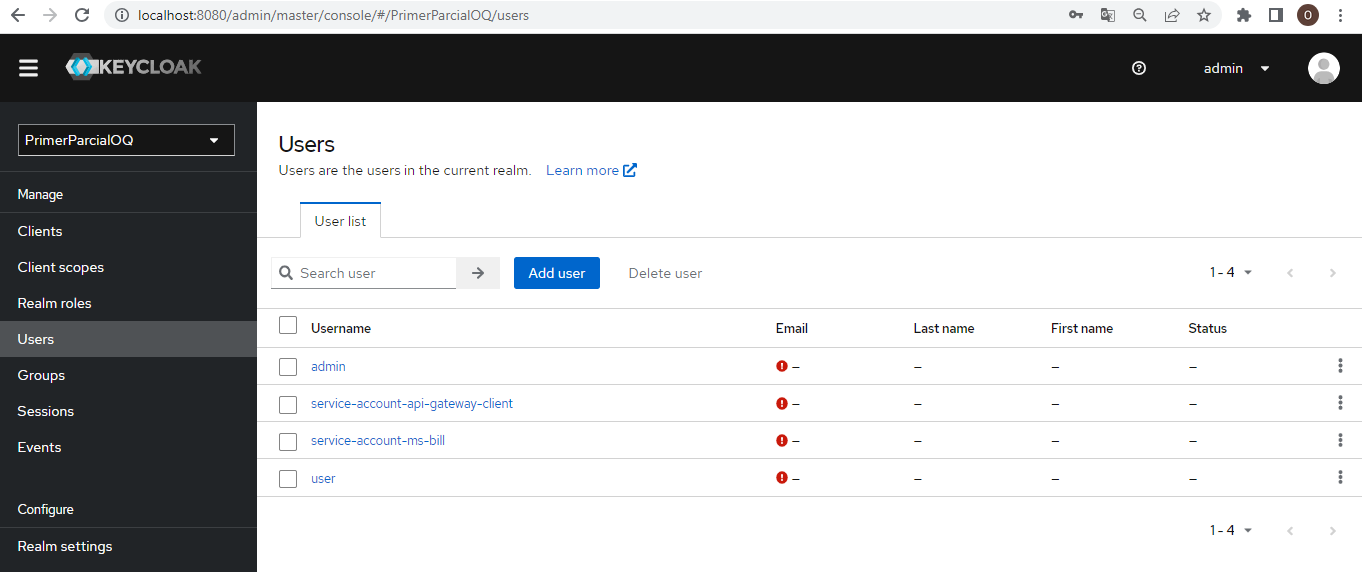


Api-gateaway-client: con dicho cliente se conectó el microservicio Gateaway y se le asignó el rol de app\_user (rol de reino) compuesto con el rol de cliente (user).

Ms-bill: con dicho cliente se conectó el microservicio de bills y se le asignó el rol de app\_user (rol de reino) compuesto con el rol de cliente (user).

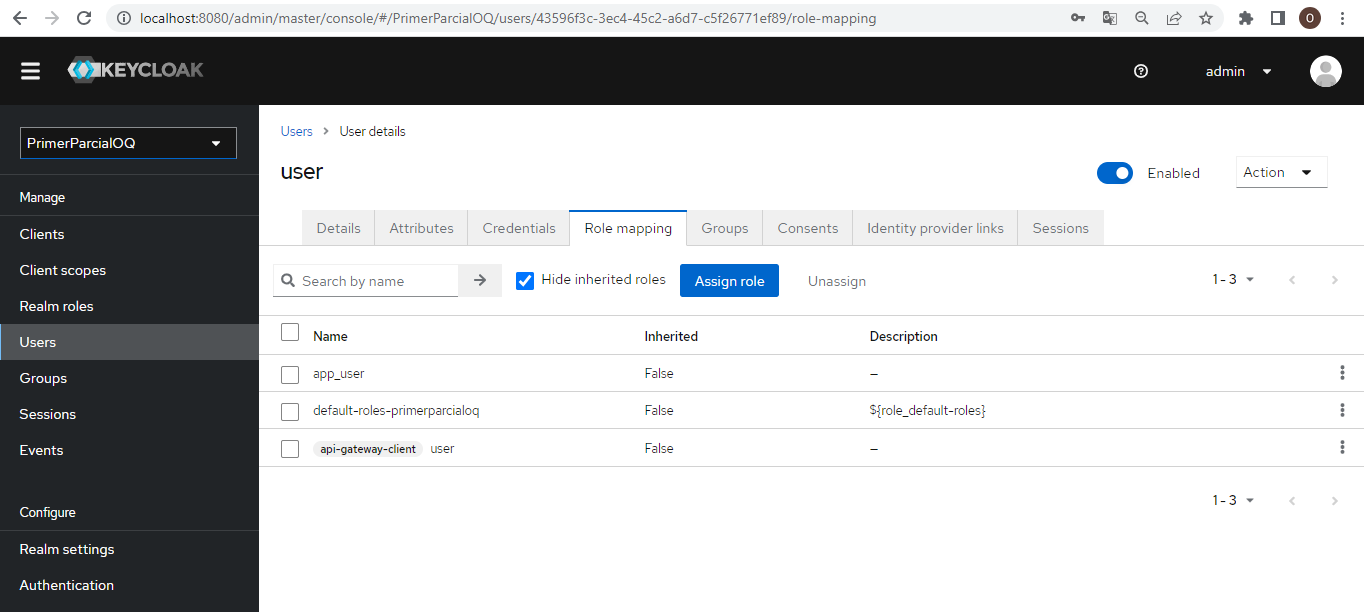
## Users

Se generaron dos usuarios (admin y user).



Ambos tienen la contraseña password123.

User tiene los roles asignados app\_user y user. Al usuario admin no se le otorgaron roles.



# Configuración del Gateway

Application.yml

server:  
 port: 9090

#Configuración de eureka discovery

eureka:  
 instance:  
 hostname: localhost  
 prefer-ip-address: true  
 client:  
 register-with-eureka: true  
 fetch-registry: true  
 serviceUrl:  
 defaultZone: http://localhost:8761/eureka

#Configuración de Spring  
spring:  
 cloud:  
 gateway:  
 default-filters:  
 - TokenRelay #filtro solicitado en la consigna  
 routes:  
 - id: ms-bill #id del microservicio  
 uri: lb://ms-bill #uri del microservicio  
 predicates:  
 - Path=/api/v1/bills/\*\* #path del microservicio  
 *#filters: StripPrefix=2* security:  
 oauth2:  
 client:  
 provider:  
 keycloak-provider:  
 issuer-uri: http://localhost:8080/realms/PrimerParcialOQ #uri de kaycloak  
 registration:  
 api-gateway-service:  
 provider: keycloak-provider  
 client-id: api-gateway-client #nombre del cliente gateway en keycloak  
 client-secret: eXYqDAYEy7DSBf5SPB7ecPF89lQqvpbV #obtenido de keycloak  
 scope: openid  
 authorization-grant-type: authorization\_code  
 redirect-uri: 'http://localhost:9090/login/oauth2/code/keycloak'

# Configuración del ms-bill

Application.properties

# Server Configuration  
server.port=${PORT:0}  
server.servlet.context-path=/api/v1/  
  
# Spring Configuration  
spring.application.name=ms-bill  
# Database  
spring.datasource.url=jdbc:h2:mem:testdb;DB\_CLOSE\_DELAY=-1  
spring.datasource.driver-class-name=org.h2.Driver  
spring.datasource.username=sa  
spring.datasource.password=sa  
# JPA  
spring.jpa.show-sql=true  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.properties.hibernate.format\_sql=true  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.H2Dialect  
spring.jpa.open-in-view=false  
# H2  
spring.h2.console.enabled= true  
spring.h2.console.path=/h2-console  
  
eureka.instance.hostname=localhost  
eureka.instance.instance-id=${spring.application.name}:${spring.application.instance\_id:${random.value}}  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka  
  
spring.security.oauth2.client.provider.keycloak.issuer-uri= http://localhost:8080/realms/PrimerParcialOQ  
spring.security.oauth2.client.registration.keycloak.client-id= ms-bills  
spring.security.oauth2.client.registration.keycloak.client-secret = QghANj1WXSS9gELgCH22iHlNiQajwOBW #obtenido de keycloak  
spring.security.oauth2.client.registration.keycloak.redirect-uri = http://localhost:9090/login/oauth2/code/keycloak  
spring.security.oauth2.resourceserver.jwt.issuer-uri= http://localhost:8080/realms/PrimerParcialOQ  
spring.security.oauth2.resourceserver.jwt.jwk-set-uri=${spring.security.oauth2.resourceserver.jwt.issuer-uri}/protocol/openid-connect/certs  
jwt.auth.converter.resource-id=ms-bills  
jwt.auth.converter.principal-attribute=preferred\_username  
logging.level.org.springframework.security=DEBUG

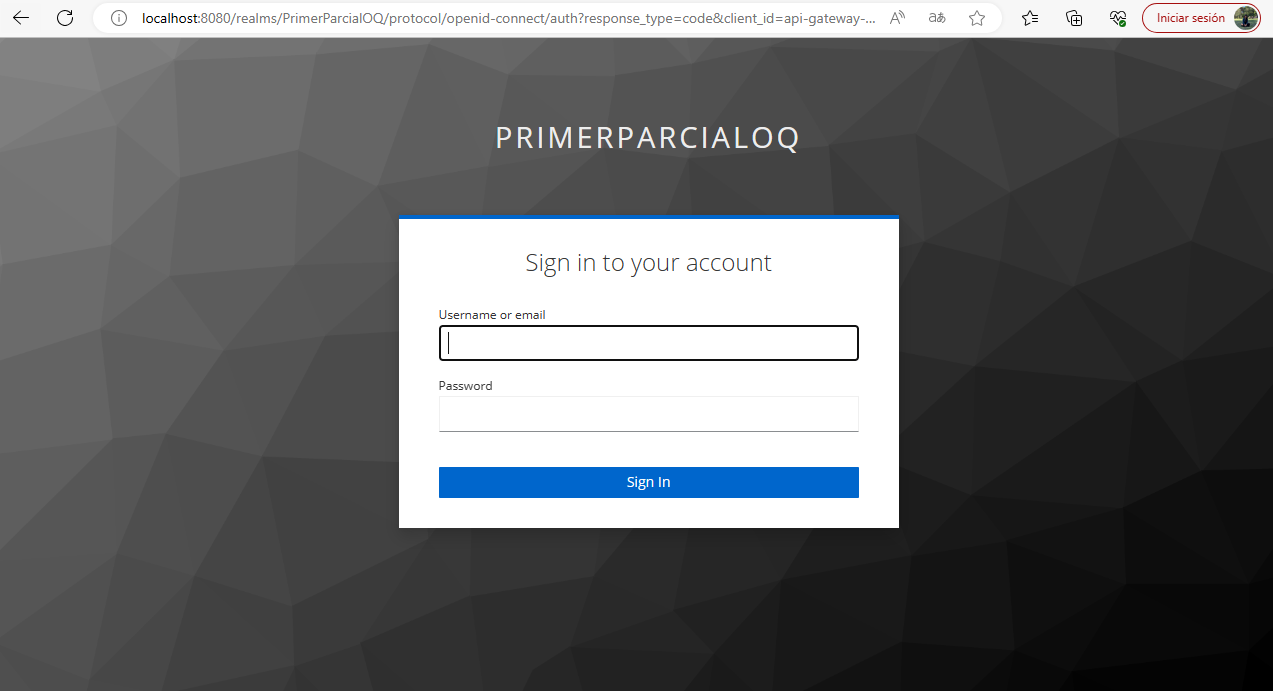
# Configuración de Eureka

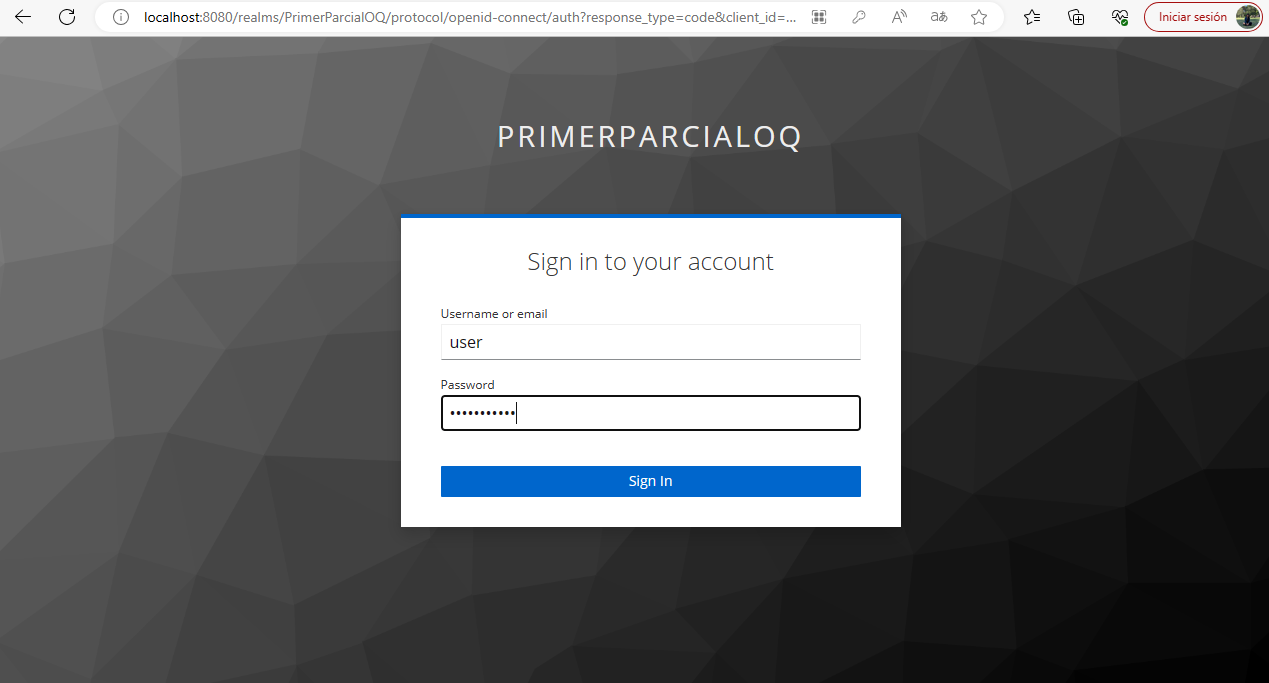
Application.properties

spring.application.name=ms-discovery  
server.port=8761  
  
eureka.instance.hostname=localhost  
eureka.instance.prefer-ip-address=true  
eureka.client.fetch-registry=false  
eureka.client.register-with-eureka=false

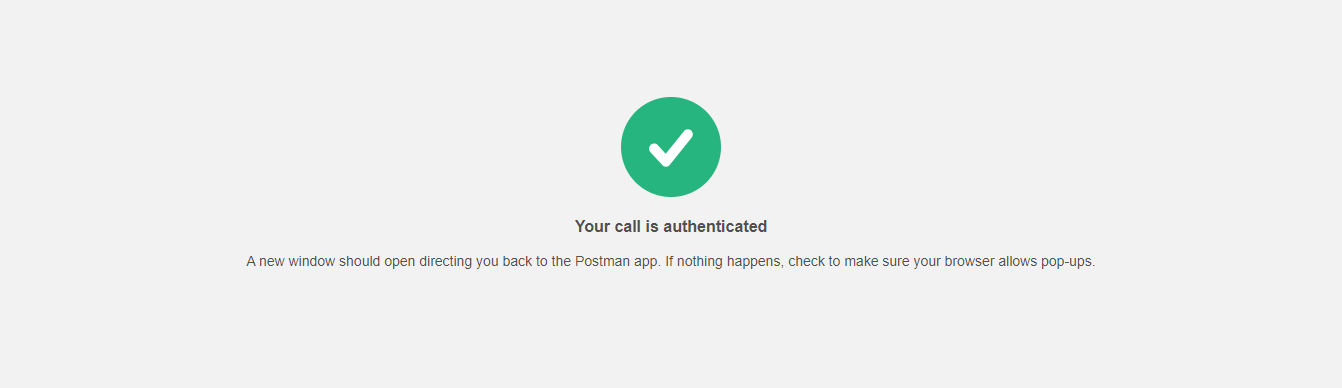
# Ingreso mediante navegador

Una vez configurado, se levantaron los 3 micro servicios y se ingresó a [localhost](http://localhost:9090/api/v1/bills/all) (localhost:9090/api/v1/bills/all) en el navegador para obtener la totalidad de las bills y redireccionó a Keycloak para autenticarse.

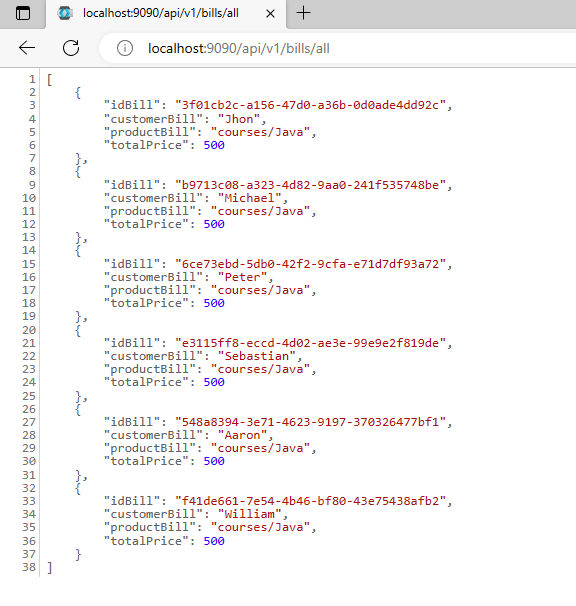




Como el usuario user tiene permisos para ingresar, la llamada fue autenticada

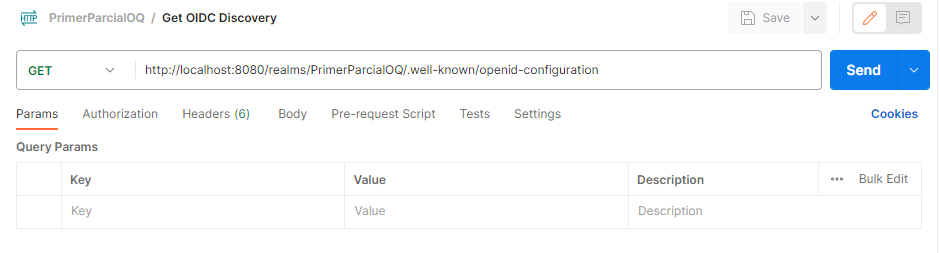


Se pudo observar las facturas en su totalidad.

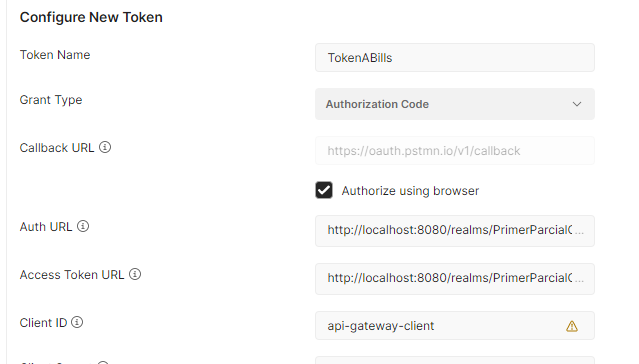


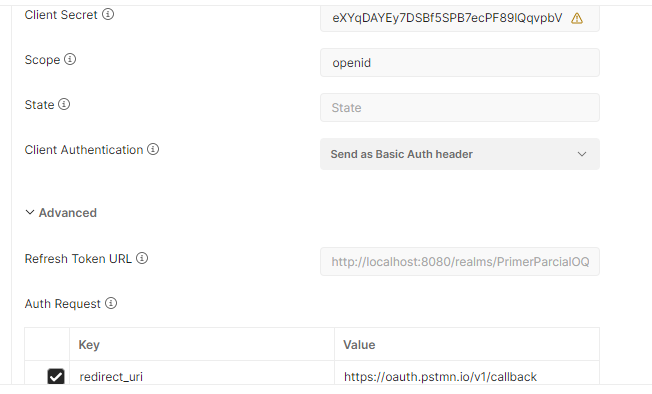
# Ingreso mediante Postman

Se realiza la consulta OIDC Discovery para obtener las urls necesarias.

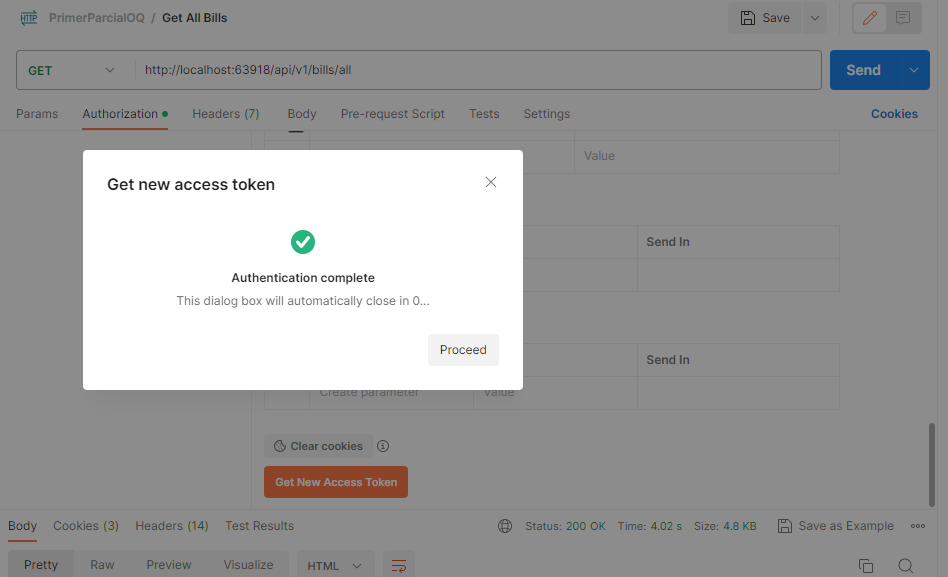


Con los endpoints obtenidos y las claves obtenidas en Keycloak se configura el new token para obtener la authorization.

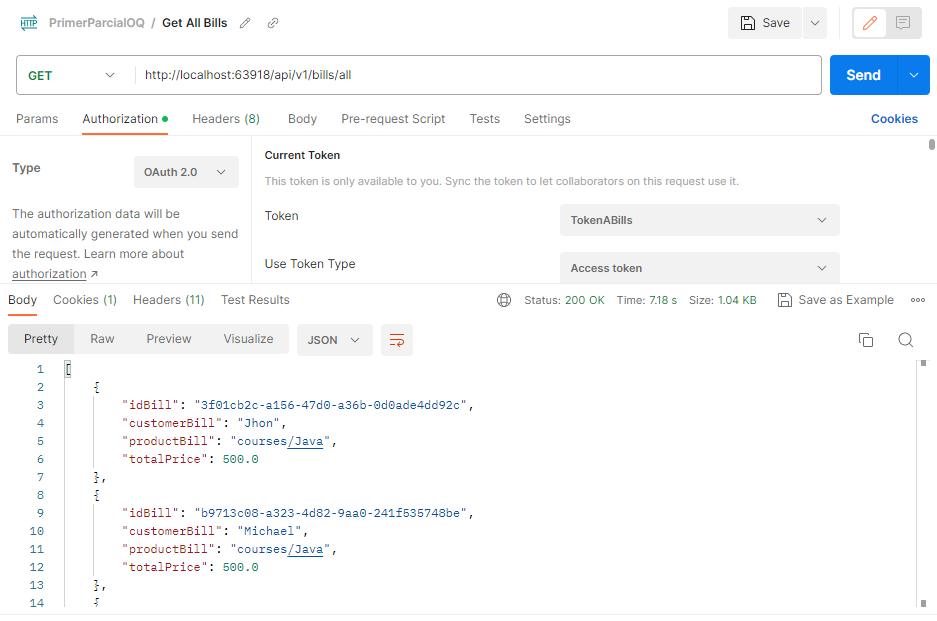




Al realizar el submit, redirecciona a keycloak para autenticarse.



Obtenemos el json con la totalidad de las bills.



# Versiones utilizadas

Versión de Keycloak 21.0.1

Versión de Springboot: 3.0.7

Versión de Java: 17

Versión de Spring Security: 3.0.7