

# BE2 - Documentación de proyecto final

## TiendaDH

**Repositorio publico:** [https://github.com/TomasMontivero/Backend2\\_Keycloak](https://github.com/TomasMontivero/Backend2_Keycloak)

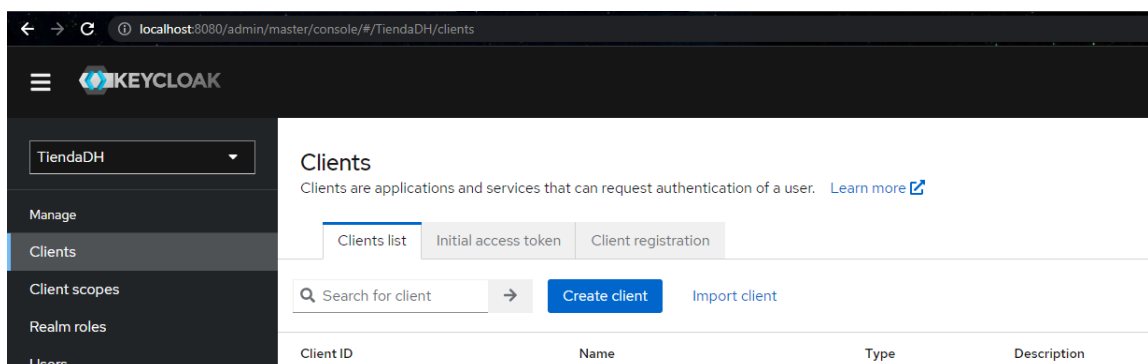
### Documentacion Online:

<https://docs.google.com/document/d/1HNHBPk6dJN83vNEhh6NtcOSWPIVfVj7KW4t5M4lb74/edit?usp=sharing>

## Ejecución del proyecto

### Ejecutar el Keycloak Runner para crear el realm TiendaDH

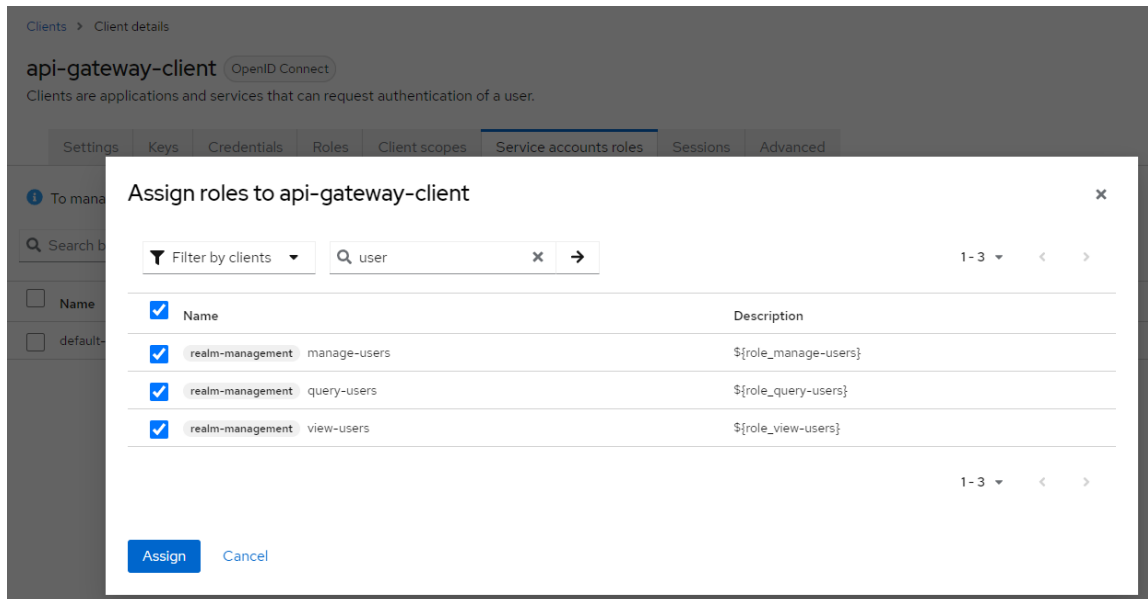
- El runner va a buscar Keycloak en el puerto 8080, y usa user:admin / password:admin  
Si se está levantando Keycloak en otro puerto o con otras credenciales, hay que modificar el application.properties:  
**keycloak-runner/src/main/resources/application.properties**
- Ir al microservicio keycloak-runner y ejecutar el KeycloakRunnerApplication  
**keycloak-runner/src/main/java/com/dh/keycloakrunner/KeycloakRunnerApplication.java**



- Seteo manual necesario para terminar de configurarlo porque no se pudo implementar en el runner:  
<http://localhost:8080/admin/master/console/#/TiendaDH/clients>

Ir a **Clientes** y entrar al **api-gateway-client**

Ir a la pestaña de **Service Accounts Roles** y asignar los roles “manage-users”, “query-users” y “view-users”



## Ejecutar Eureka

- Ir al microservicio ms-discovery y ejecutar el MsDiscoveryApplication  
**ms-discovery/src/main/java/com/dh/msdiscovery/MsDiscoveryApplication.java**



## Ejecutar Gateway

- Ir al microservicio ms-gateway y ejecutar el MsGatewayApplication  
**ms-gateway/src/main/java/com/dh/msgateway/MsGatewayApplication.java**

## Ejecutar Bills

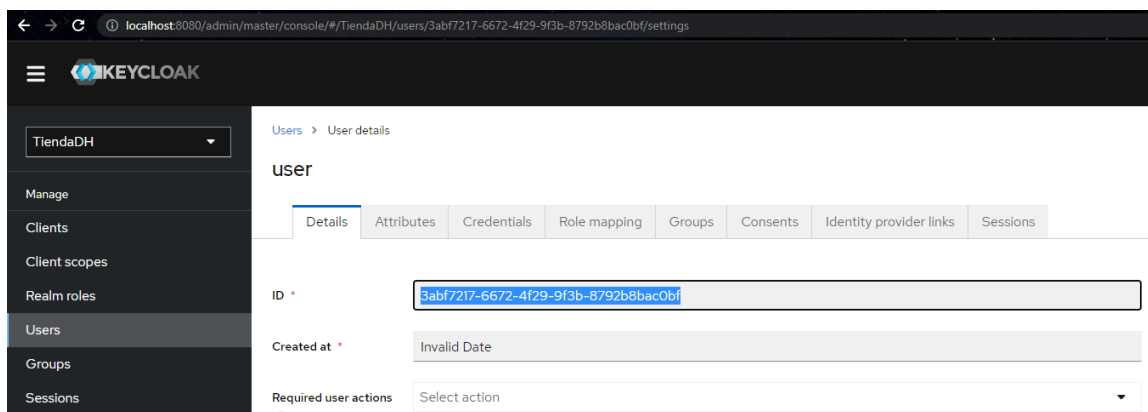
- Ir al microservicio ms-bills y ejecutar el MsBillsApplication  
**ms-bills/src/main/java/com/dh/msbills/MsBillsApplication.java**

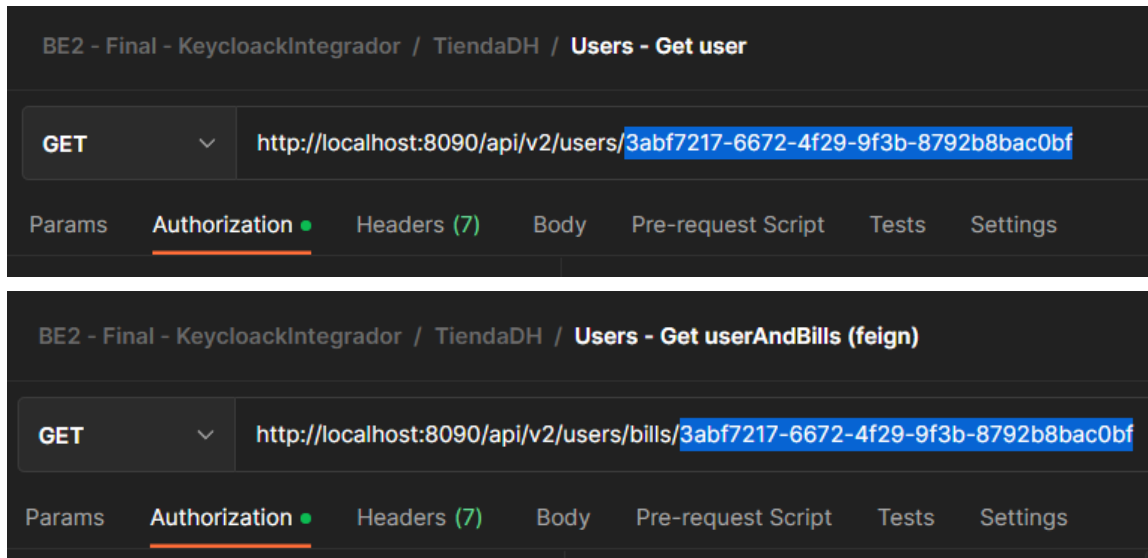
## Ejecutar Users

- Ir al microservicio ms-users y ejecutar el MsUsersApplication  
**ms-users/src/main/java/com/dh/msusers/MsUsersApplication.java**

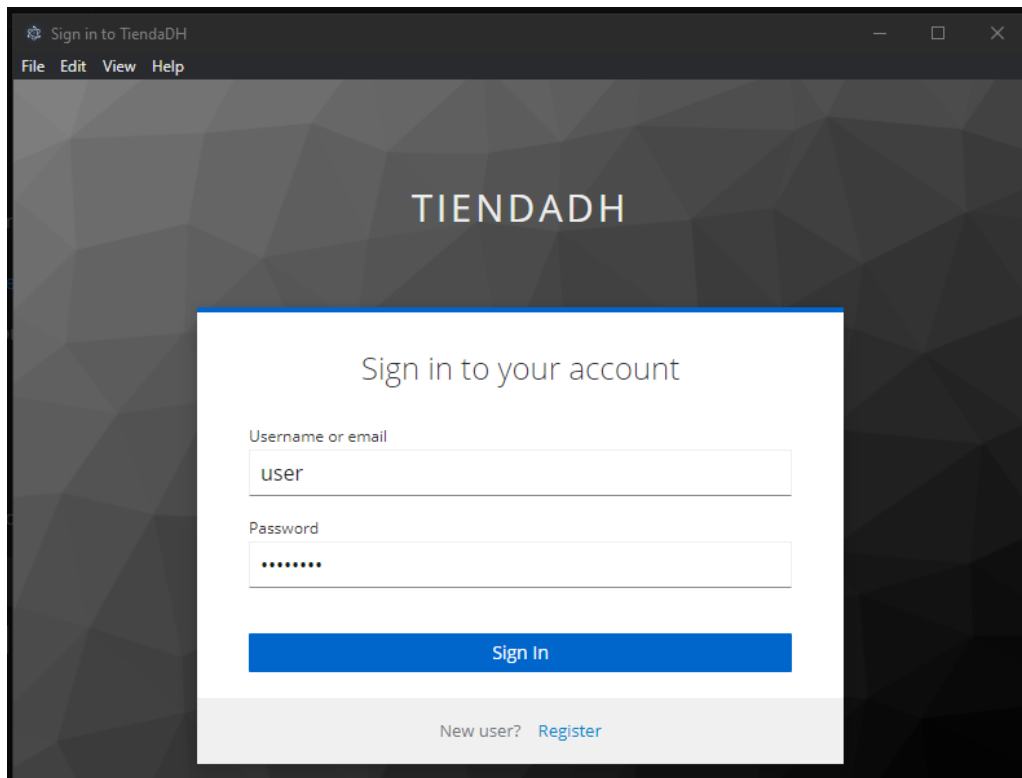
## Pruebas en Postman

- Configuración inicial
- Importar la collection en Postman
- Todos los request a Users y Bills tienen implementado el “Generate New Access Token” en la pestaña de Authorization
- Seteo manual necesario:  
En los 2 GET de Users, es necesario insertar en la URL el ID del user “user” que se generó en Keycloak



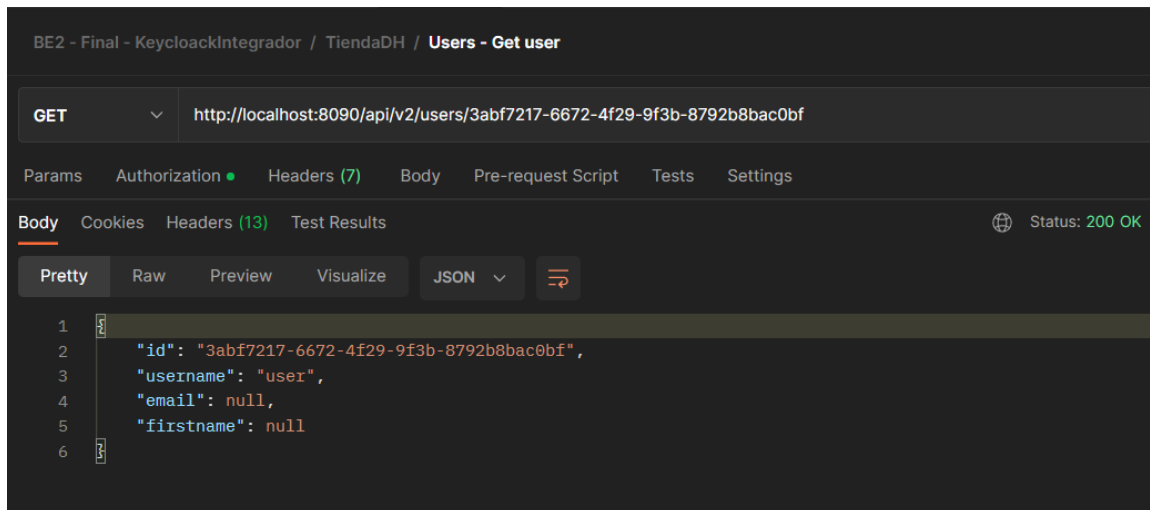


- Haciendo click en “Generate New Access Token” se debe abrir una ventana de navegador para iniciar sesion en Keycloak.  
Usar las credenciales: user: **user** // password: **password**



- Users - Get user

El endpoint `/users/{id}` devuelve la información del usuario que obtiene de Keycloak



```
BE2 - Final - KeycloakIntegrador / TiendaDH / Users - Get user

GET http://localhost:8090/api/v2/users/3abf7217-6672-4f29-9f3b-8792b8bac0bf

Params Authorization Headers (7) Body Pre-request Script Tests Settings

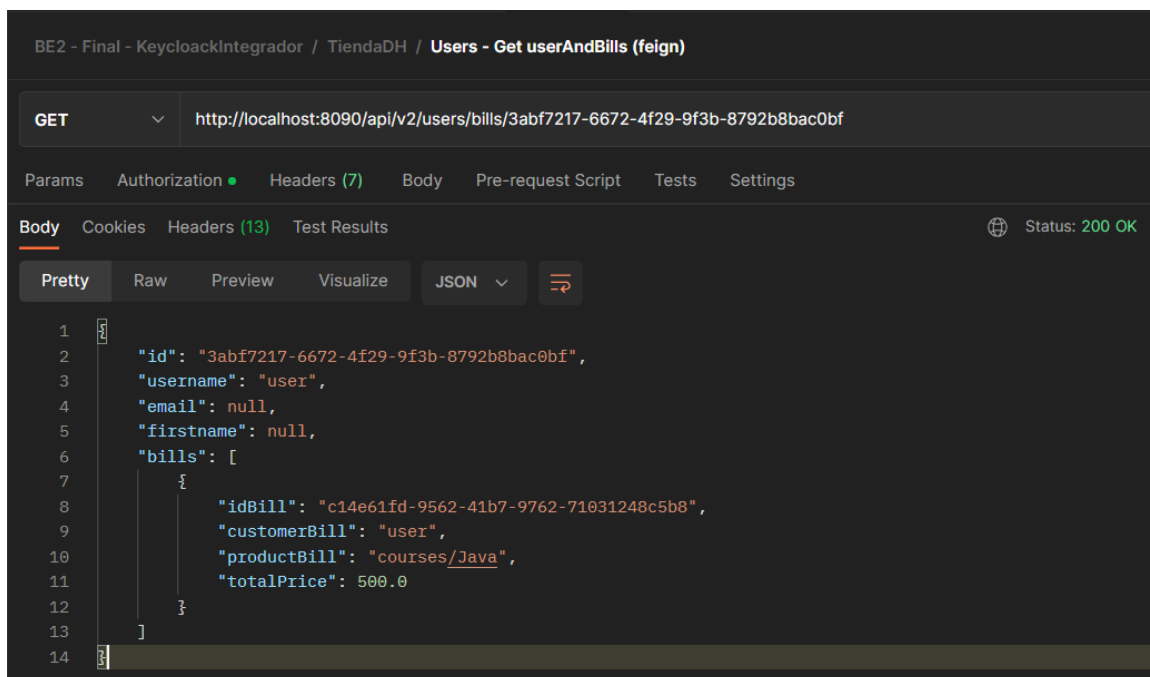
Body Cookies Headers (13) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

1 {
2   "id": "3abf7217-6672-4f29-9f3b-8792b8bac0bf",
3   "username": "user",
4   "email": null,
5   "firstname": null
6 }
```

- Users - Get user and bills (feign)

El endpoint `/users/bills/{id}` devuelve la misma información del usuario y adiciona las facturas que obtiene de Bills



```
BE2 - Final - KeycloakIntegrador / TiendaDH / Users - Get userAndBills (feign)

GET http://localhost:8090/api/v2/users/bills/3abf7217-6672-4f29-9f3b-8792b8bac0bf

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Body Cookies Headers (13) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

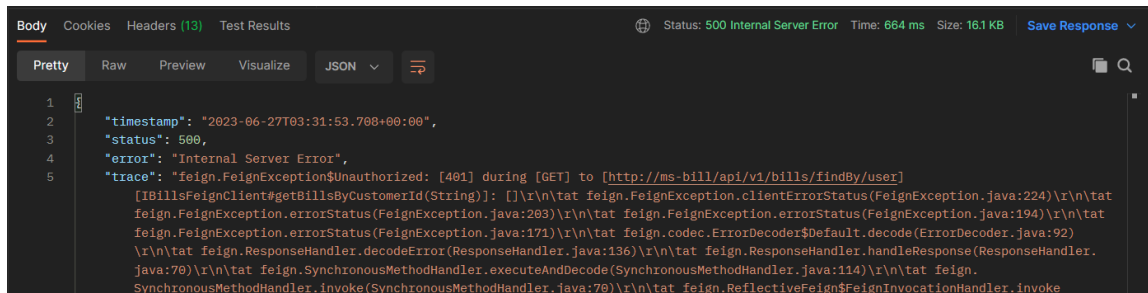
1 {
2   "id": "3abf7217-6672-4f29-9f3b-8792b8bac0bf",
3   "username": "user",
4   "email": null,
5   "firstname": null,
6   "bills": [
7     {
8       "idBill": "c14e61fd-9562-41b7-9762-71031248c5b8",
9       "customerBill": "user",
10      "productBill": "courses/Java",
11      "totalPrice": 500.0
12    }
13  ]
14 }
```

**OBSERVACIÓN:**

Después de unos minutos de levantar el ms-users, el endpoint empieza a devolver permanentemente un 401 al comunicarse con msbills

Incluso generando un nuevo token, internamente parece no renovarlo. No logré encontrar en el código por que falla.

**SOLUCIÓN:** Apagar y volver a iniciar el microservicio de Users desde IntelliJ, esperar un minuto y volver a probar el endpoint.



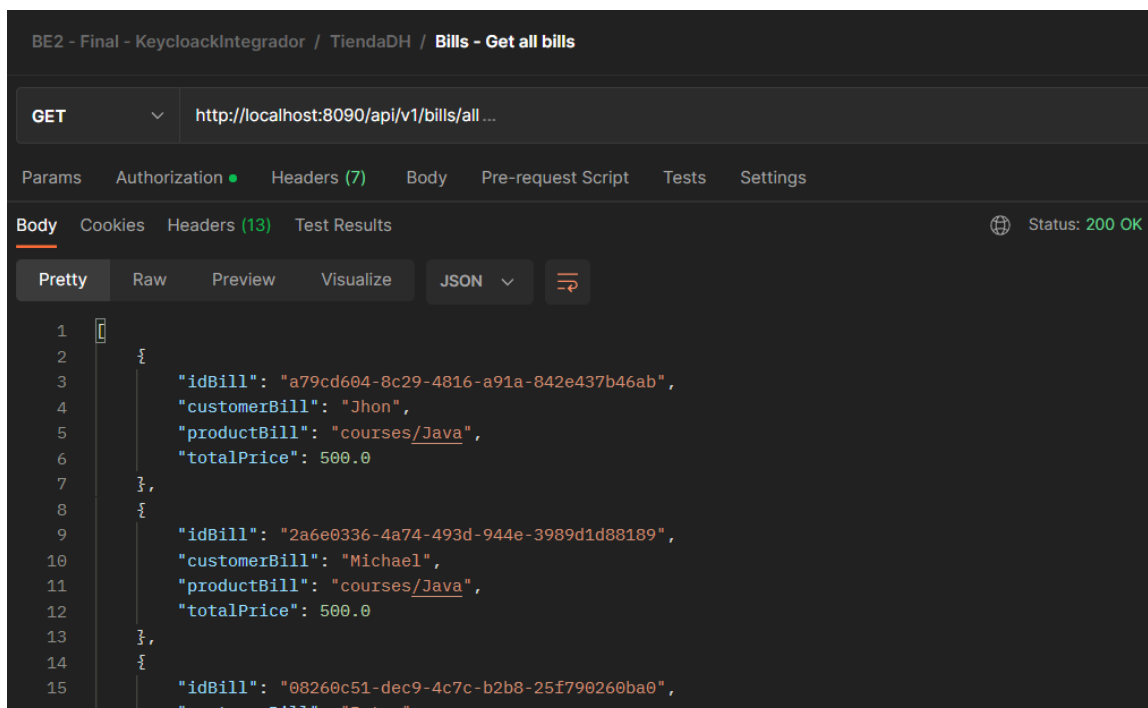
The screenshot shows a REST client interface with the following details:

- Status: 500 Internal Server Error
- Time: 664 ms
- Size: 16.1 KB
- Save Response
- Body tab selected, Pretty view, JSON format.
- Response body (JSON):

```
{  "timestamp": "2023-06-27T03:31:53.708+00:00",  "status": 500,  "error": "Internal Server Error",  "trace": "feign.FeignException$Unauthorized: [401] during [GET] to [http://ms-bill/api/v1/bills/findBy/user] [IBillsFeignClient#getBillsByCustomerId(String)]: []\r\n\tat feign.FeignException.clientErrorStatus(FeignException.java:224)\r\n\tat feign.FeignException.errorStatus(FeignException.java:203)\r\n\tat feign.FeignException.errorStatus(FeignException.java:194)\r\n\tat feign.FeignException.errorStatus(FeignException.java:171)\r\n\tat feign.codec.ErrorDecoder$Default.decode(ErrorDecoder.java:92)\r\n\tat feign.ResponseHandler.decodeError(ResponseHandler.java:136)\r\n\tat feign.ResponseHandler.handleResponse(ResponseHandler.java:70)\r\n\tat feign.SynchronousMethodHandler.executeAndDecode(SynchronousMethodHandler.java:114)\r\n\tat feign.SynchronousMethodHandler.invoke(SynchronousMethodHandler.java:70)\r\n\tat feign.ReflectiveFeign$FeignInvocationHandler.invoke"}
```

- Bills - Get all bills

El endpoint **/bills/all** devuelve la lista completa de facturas que obtiene de su base de datos



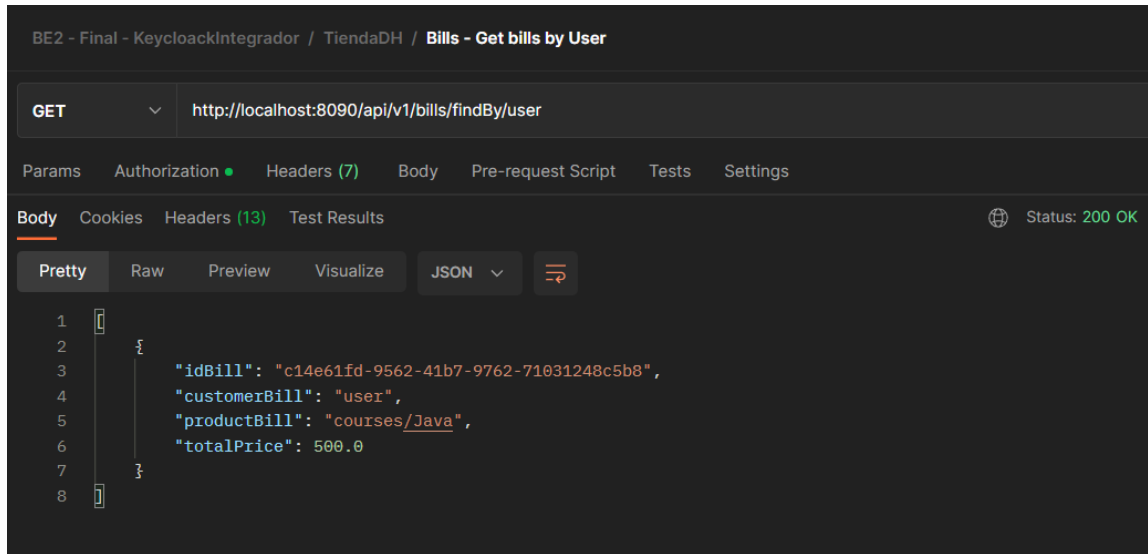
The screenshot shows a REST client interface with the following details:

- URL: http://localhost:8090/api/v1/bills/all ...
- Method: GET
- Status: 200 OK
- Body tab selected, Pretty view, JSON format.
- Response body (JSON):

```
{  "idBill": "a79cd604-8c29-4816-a91a-842e437b46ab",  "customerBill": "Jhon",  "productBill": "courses/Java",  "totalPrice": 500.0}, {  "idBill": "2a6e0336-4a74-493d-944e-3989d1d88189",  "customerBill": "Michael",  "productBill": "courses/Java",  "totalPrice": 500.0}, {  "idBill": "08260c51-dec9-4c7c-b2b8-25f790260ba0",  "customerBill": "Peter"}
```

- Bills - Get bills by user

El endpoint **/bills/findBy/{customerBill}** devuelve las facturas que haya encontrado para el usuario pasado por url

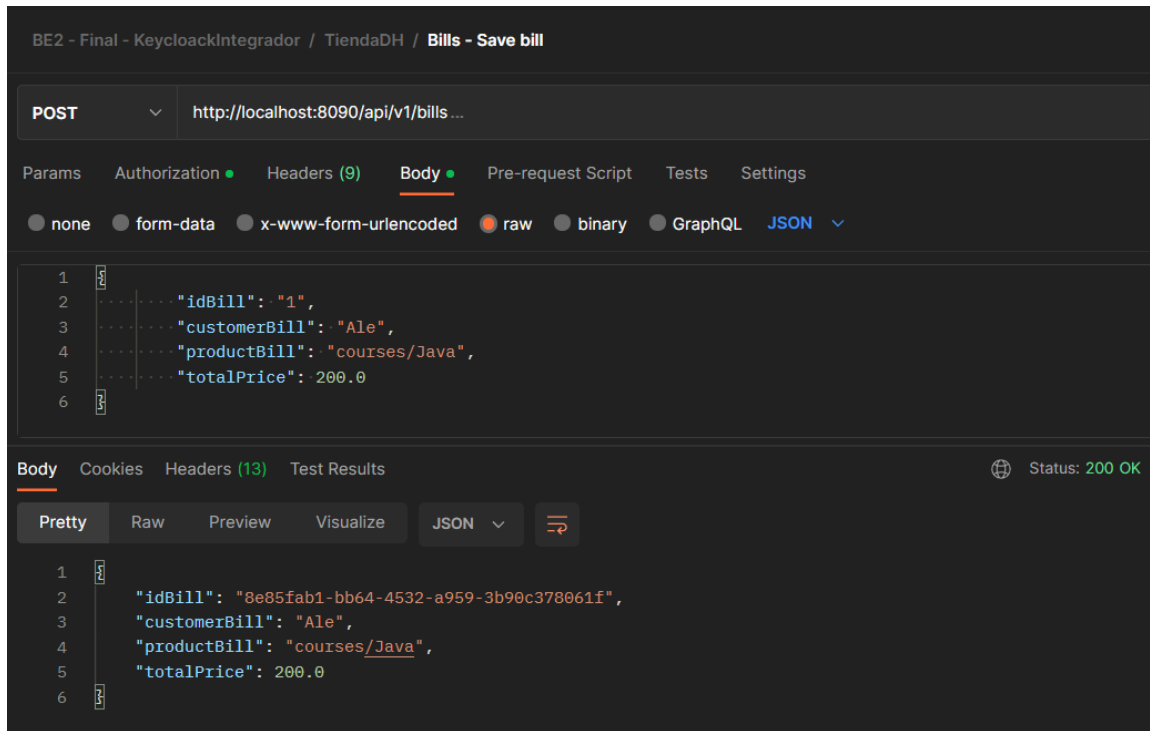


- Bills - Save bill

El endpoint **/bills** recibe por body el Json de una factura y la guarda en la base de datos.

Borrando cookies de postman y generando un nuevo access token con las credenciales:  
user: **provider** // password: **password**

El endpoint devuelve 200 y un Json con la misma factura que se guardó. Porque el usuario provider pertenece al grupo "PROVIDERS"



Si se usa el POST con el token del usuario “user” (password: password) el endpoint devuelve 403 porque el usuario no pertenece al grupo “PROVIDERS

