

## DIGITAL HOUSE

1er trabajo evaluador

EB2 – Camada 1

Sonia Gisela Urriza y Spreafichi

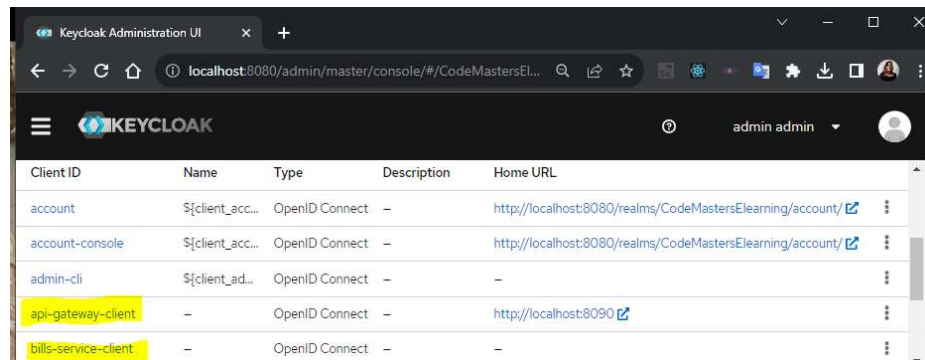
Documentación respaldatoria

### CONFIGURACIONES KEYCLOAK



## Welcome to CodeMastersElearning

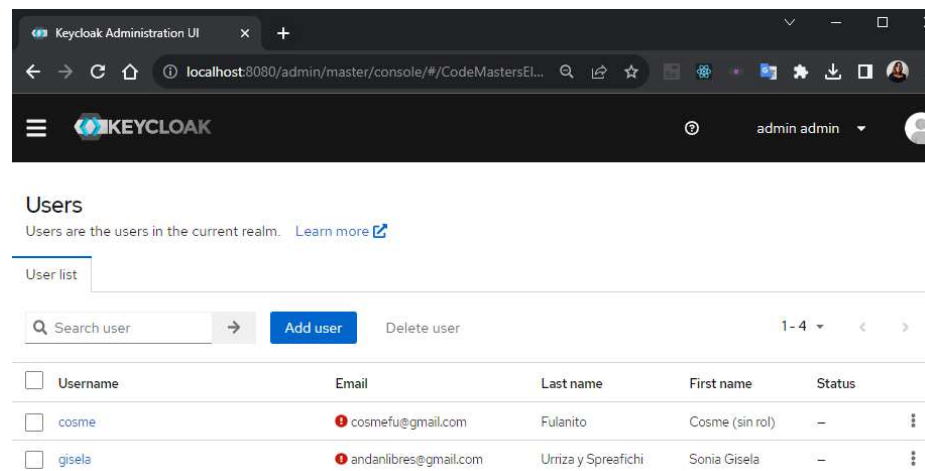
Clientes:



The screenshot shows the Keycloak Administration UI with the 'Clients' tab selected. The table lists several clients, with 'api-gateway-client' and 'bills-service-client' highlighted in yellow.

Client ID	Name	Type	Description	Home URL
account	\$client_acc...	OpenID Connect	-	<a href="http://localhost:8080/realms/CodeMastersElearning/account/">http://localhost:8080/realms/CodeMastersElearning/account/</a>
account-console	\$client_acc...	OpenID Connect	-	<a href="http://localhost:8080/realms/CodeMastersElearning/account/">http://localhost:8080/realms/CodeMastersElearning/account/</a>
admin-cli	\$client_ad...	OpenID Connect	-	-
api-gateway-client	-	OpenID Connect	-	<a href="http://localhost:8090/">http://localhost:8090/</a>
bills-service-client	-	OpenID Connect	-	-

Users:



The screenshot shows the Keycloak Administration UI with the 'Users' tab selected. The 'User list' section displays a table of users, including 'cosme' and 'gisela'.

Users

Users are the users in the current realm. [Learn more](#)

User list

Search user → Add user Delete user 1 - 4 < >

<input type="checkbox"/>	Username	Email	Last name	First name	Status
<input type="checkbox"/>	cosme	cosmefu@gmail.com	Fulanito	Cosme (sin rol)	-
<input type="checkbox"/>	gisela	andanlibres@gmail.com	Urriza y Spreafichi	Sonia Gisela	-

api-gateway-client

Access settings

Root URL ⓘ

http://localhost:8090

Home URL ⓘ

Valid redirect URIs ⓘ

http://localhost:8090/\*

Add valid redirect URIs

Valid post logout redirect URIs ⓘ

Add valid post logout redirect URIs

Web origins ⓘ

http://localhost:8090

Add web origins

Admin URL ⓘ

http://localhost:8090

Capability config

Client authentication ⓘ

On

Authorization ⓘ

Off

Authentication flow

Standard flow ⓘ

Direct access grants ⓘ

Implicit flow ⓘ

Service accounts roles ⓘ

Bills-service-client

Access settings

Root URL ⓘ

Home URL ⓘ

Valid redirect URIs ⓘ

/\*

Add valid redirect URIs

Valid post logout redirect URIs ⓘ

Add valid post logout redirect URIs

Web origins ⓘ

/\*

Add web origins

Admin URL ⓘ

Capability config

Client authentication ⓘ

On

Authorization ⓘ

Off

Authentication flow

Standard flow ⓘ

Direct access grants ⓘ

Implicit flow ⓘ

Service accounts roles ⓘ

Realm Roles:

Keycloak Administration UI

localhost:8080/admin/...

admin admin

Realm roles

Realm roles are the roles that you define for use in the current realm. [Learn more](#)

Search role by name

Create role

1 - 4

Role name	Composite	Description
app_user	True	-
default-roles-codemasterselearning ⓘ	True	`\${role_default-roles}`

Keycloak Administration UI

localhost:8080/admin/...

admin admin

DetailsAssociated rolesAttributesUsers in role

Search by name

Hide inherited roles

Assign role

Unassign

1 - 2

Name	Inherited	Description
api-gateway-client user	False	-
bills-service-client user	False	-

Keycloak Administration UI

localhost:8080/admin/...

admin admin

Realm roles > Role details

app\_userCompositeAction

DetailsAssociated rolesAttributesUsers in role

Who will appear in this user list?

1 - 1

Username	Email	Last name	First name
gisela	andanlibres@gmail.com	Urriza y Spreafichi	Sonia Gisela

## Scope:

The first screenshot shows the 'Client scopes' page in the Keycloak Administration UI. It lists various built-in scopes like 'acr', 'address', 'email', 'microprofile-jwt', 'offline\_access', 'phone', and 'print'. The 'print' scope is highlighted in yellow.

The second screenshot shows the 'Client details' page for the 'api-gateway-client'. The 'Client scopes' tab is selected, showing a list of assigned scopes. The 'print' scope is highlighted in yellow.

The third screenshot shows the 'Client scope details' page for the 'print' scope. The 'Scope' tab is selected, showing a message about role scope mapping and a table of roles. The 'app\_user' role is listed with 'Inherited' set to 'False'.

Name	Assigned type	Protocol	Displa...	Description
acr	Default	OpenID Connect	—	OpenID Connect scope for add acr (auth...
address	Optional	OpenID Connect	—	OpenID Connect built-in scope: address
email	Default	OpenID Connect	—	OpenID Connect built-in scope: email
microprofile-jwt	Optional	OpenID Connect	—	Microprofile - JWT built-in scope
offline_access	Optional	OpenID Connect	—	OpenID Connect built-in scope: offline_acc
phone	Optional	OpenID Connect	—	OpenID Connect built-in scope: phone
print	Optional	OpenID Connect	—	—

Assigned client scope	Assigned type	Description
api-gateway-client-dedicated	none	Dedicated scope and mappers for this client
acr	Default	OpenID Connect scope for add acr (authentication context class reference) to the token
address	Optional	OpenID Connect built-in scope: address
email	Default	OpenID Connect built-in scope: email
microprofile-jwt	Optional	Microprofile - JWT built-in scope
offline_access	Optional	OpenID Connect built-in scope: offline_access
phone	Optional	OpenID Connect built-in scope: phone
print	Optional	—
profile	Default	OpenID Connect built-in scope: profile
roles	Default	OpenID Connect scope for add user roles to the access token

Name	Inherited	Description
app_user	False	—

Con **Role Scope Mapping** definido. Sólo usuarios con rol app\_user, o en este caso rol user (ya que el rol app\_user está asociado al rol de cliente user), podrán acceder al Scope "print".

## CONFIGURACION MICROSERVICIOS

### VERSIONES:

- Spring Boot: 3.1.2
- SDK: 17

### LEVANTAR MICROSERVICIOS – PASOS:

1. Docker Desktop
2. Config-server
3. Ms-discovery
4. Ms-bills
5. Ms-gateway

Eureka: 8761  
Gateway: 8090  
Ms-bills: aleatorio

Facturación: <http://localhost:8090/api/v1/bills/all>

Eureka: <http://localhost:8761/>

Reino - OpenID Endpoint Configuration: <http://localhost:8080/realms/CodeMastersElearning/.well-known/openid-configuration>

```
// PERSONALIZADO
private static List<GrantedAuthority> extractALL(String route, JsonNode jwt) {

    Set<String> allWithPrefix = new HashSet<>();

    switch (route) {
        case "resource_access":
            jwt
                .path("resource_access")
                .elements()
                .forEachRemaining(e -> e.path("roles")
                    .elements()
                    .forEachRemaining(r -> allWithPrefix.add("ROLE_" +
r.asText())));
            break;

        case "realm_access":
            jwt
                .path("realm_access")
                .path("roles")
                .elements()
                .forEachRemaining(r -> allWithPrefix.add("ROLE_" + r.asText()));
            break;

        case "aud" :
            jwt
                .path("aud")
                .elements()
                .forEachRemaining(e ->allWithPrefix.add("AUD_" + e.asText()));
            break;

        case "scope" :
            jwt
                .path("scope")
                .elements()
                .forEachRemaining(e -> allWithPrefix.add("SCOPE_" + e.asText()));
            break;

        default:
    }

    final List<GrantedAuthority> authorityList =
AuthorityUtils.createAuthorityList(allWithPrefix.toArray(new String[0]));

    return authorityList;
}
```

[illegible]

The screenshot shows a web browser window with the address bar displaying 'localhost:8090/api/v1/bills/all'. The page content is in Spanish, indicating a 403 Forbidden error: 'Se ha denegado el acceso a localhost' (Access to localhost has been denied) and 'No tienes autorización para ver esta página.' (You do not have authorization to view this page). Below this, it says 'HTTP ERROR 403'. At the bottom, there is a blue button labeled 'Volver a cargar' (Reload).

