

DIGITAL HOUSE

1er trabajo evaluador

EB2 – Camada 1

Sonia Gisela Urriza y Spreafichi

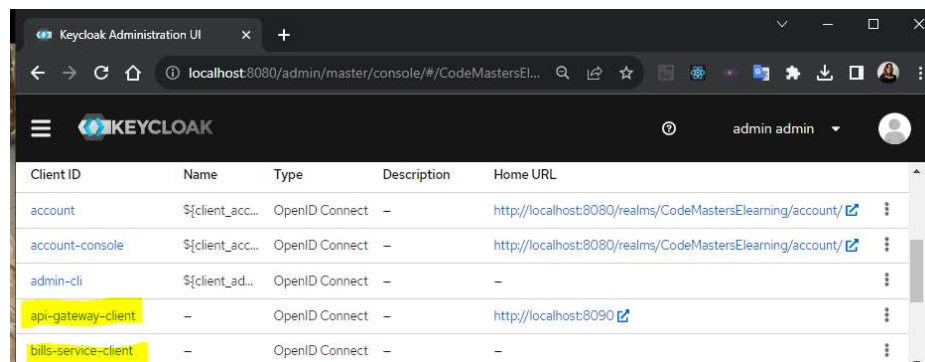
Documentación respaldatoria

CONFIGURACIONES KEYCLOAK



Welcome to CodeMastersElearning

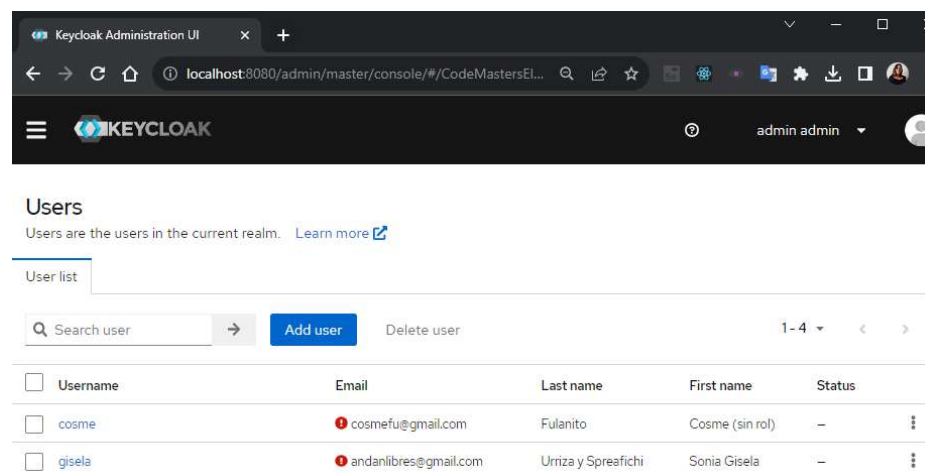
Clientes:



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	-	http://localhost:8080/realms/CodeMastersElearning/account/
account-console	\$client_acc...	OpenID Connect	-	http://localhost:8080/realms/CodeMastersElearning/account/
admin-cli	\$client_ad...	OpenID Connect	-	-
api-gateway-client	-	OpenID Connect	-	http://localhost:8090
bills-service-client	-	OpenID Connect	-	-

Users:



The screenshot shows the Keycloak Administration UI with the 'Users' tab selected. The table lists two users: 'cosme' and 'gisela'.

Username	Email	Last name	First name	Status
cosme	cosmefu@gmail.com	Fulanito	Cosme (sin rol)	-
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

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

PUERTOS:

Eureka: 8761

Gateway: 8090

Ms-bills: aleatorio

LINKS

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>

JWT CONVERTER CUSTOMIZADO (SWITCH)

```
private static Collection<? extends GrantedAuthority> extractResourceRoles(final Jwt
jwt) throws JsonProcessingException {
    Set<GrantedAuthority> resourcesRoles = new HashSet();
    ObjectMapper objectMapper = new ObjectMapper();
    objectMapper.registerModule(new JavaTimeModule());

    resourcesRoles.addAll(extractALL("resource_access",
objectMapper.readTree(objectMapper.writeValueAsString(jwt)).get("claims")));
    resourcesRoles.addAll(extractALL("realm_access",
objectMapper.readTree(objectMapper.writeValueAsString(jwt)).get("claims")));

    return resourcesRoles;
}
// PERSONALIZANDO EXTRACCIÓN. Es más interesante cuando hay más factores.
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()));

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

    return authorityList;
}
```

APLICACIÓN EN FUNCIONAMIENTO

USUARIO CON ROL	USUARIO SIN ROL
usuario: gisela	usuario: cosme
contraseña: password	contraseña: password
Listado de facturación http://localhost:8090/api/v1/bills/all	

