# LAB 15: QUARKUS SECURE REVIEW

Autor: José Díaz

Github Repo: https://github.com/joedayz/quarkus-bcp-2025.git

Abre el proyecto **13-secure-review-start**.

### Instrucciones

Este ejercicio usa la aplicación speaker como back end. El back end se integra con un servidor Keycloak para autenticación y autorización. Adicionalmente, el backend se integra con una aplicación front-end SPA.

1. Abre la aplicación expenses

1.1. Navega al directorio 13-secure-review
1.2. Abre el proyecto con tu editor favorito.

2. Integra la aplicación speaker con el servidor SSO. Usa la siguiente configuración:
   - SSO Server URL: http://localhost:8888
   - Keycloak realm: quarkus
   - Client Id: backend-service
   - Client secret: secret

3. Configura CORS para la aplicación speaker. La aplicación debería permitir solo requests del origen localhost en el puerto 9000 en dev u 8080 en prod. Denegar requests de otros origenes.
4. Configurar la autorización de los siguientes endpoints:
   - GET /speakers: necesita el rol de read.
   - GET /speakers/{uuid}: necesita el rol de read.
   - POST /speakers: necesita el rol de modify.
   - PUT /speakers/{uuid}: necesita el rol de modify.
5. Opcionalmente, usa el front-end speaker-dashboard para probar la aplicación speaker.
5.1. Inicia el servicio speaker: **mvn quarkus:dev**
5.2. En un navegador web, por ejemplo, Firefox, abre el http://localhost:8888 para validar que el Keycloak esta corriendo sin problemas. Sino en speaker service levanta el servicio con docker compose up -d | podman compose up -d.

   NOTA IMPORTANTE: Se ha configurado la aplicación frontend-service en el realm.json

5.3. Abre el http://localhost:9000. Usa el usuario user y password redhat. Tu debes ver un dashboard con 4 speakers.
5.4. Click Add a speaker. Ingresa tus nombres y apellidos en first name y last name respectivamente. Luego, haz click en Confirm. Se te mostrará un error, porque no estas autorizado a crear speakers. Cierra todas las ventanas para deslogueate.
5.5. En una nueva ventana, abre http://localhost:9000. Usa el usuario superuser y password redhat.
5.6. Click en Add a speaker. Ingresa tus nombres y apellidos nuevamente en first name y last name respectivamente. Luego click en Confirm. La llamada funcionará satisfactoriamente,

porque el usuario superuser si puede crear speakers. Cierra la ventana del navegador.

5.7. Retorna a la terminal y ejecuta el servicio speaker y presiona la letra q para detener la aplicación.

**Solución:**

Este ejercicio usa la aplicación speaker como back end. El back end se integra con un servidor Keycloak para autenticación y autorización. Adicionalmente, el backend se integra con una aplicación front-end SPA.

1. Abre la aplicación expenses

1.1. Navega al directorio 13-secure-review
1.2. Abre el proyecto con tu editor favorito.

2. Integra la aplicación speaker con el servidor SSO. Usa la siguiente configuración:
   - SSO Server URL: http://localhost:8888
   - Keycloak realm: quarkus
   - Client Id: backend-service
   - Client secret: secret

2.1. Agrega la extension quarkus-oidc al proyecto: **mvn quarkus:add-extension -Dextensions=oidc**

2.2. Configura la integración con OIDC agregando las siguientes propiedades en src/main/resources/application.properties

```
# RHSSO settings
quarkus.oidc.auth-server-url=http://localhost:8888/realms/quarkus
quarkus.oidc.client-id=backend-service
quarkus.oidc.credentials.secret=secret
quarkus.oidc.tls.verification=none
```

2.3. Verifica que el test ConfigTest pasa y soluciona cualquier problema que encuentres: **mvn clean test -Dtest=ConfigTest**

3. Configura CORS para la aplicación speaker. La aplicación debería permitir solo requests del origin localhost en el puerto 9000 y 8080. Denegar requests de otros origins.

3.1. Agrega las siguientes propiedades en src/main/resources/application.properties

```
# CORS settings
quarkus.http.cors=true
quarkus.http.cors.origins=http://localhost:9000,http://localhost:8080
quarkus.http.cors.methods=GET,POST,PUT,DELETE,OPTIONS
quarkus.http.cors.headers=accept,authorization,content-type,x-requested-with
quarkus.http.cors.exposed-headers=Content-Disposition
quarkus.http.cors.access-control-max-age=24H
```

3.2. Verifica que el CorsTest pasa: **mvn clean test -Dtest=CorsTest**

4. Configurar la autorización de los siguientes endpoints:

- GET /speakers: necesita el rol de read.
- GET /speakers/{uuid}: necesita el rol de read.
- POST /speakers: necesita el rol de modify.
- PUT /speakers/{uuid}: necesita el rol de modify.

4.1. Abre com.bcp.training.SpeakerResource y usa la anotación @RolesAllowed para asegurar los endpoints.

```
...code omitted...

@GET
@RolesAllowed("read")
public List<Speaker> getSpeakers() {

...code omitted...


@GET
@Path("/{uuid}")
@RolesAllowed("read")
public Optional<Speaker> findByUuid(@PathParam("uuid") String uuid) {

...code omitted...

@Transactional
@POST
@RolesAllowed("modify")
public Speaker insert(Speaker speaker) {

...code omitted...

@Transactional
@PUT
@Path("/{uuid}")
@RolesAllowed("modify")
public Speaker update(@PathParam("uuid") String uuid, Speaker speaker) {

...code omitted...
```

4.2. Verifica que el test SpeakerResourceTest pasa: **mvn clean test -Dtest=SpeakerResourceTest**

5.Opcionalmente, usa el front end speaker-dashboard para probar la aplicación speaker.

5.1. Inicia el speaker service: **mvn quarkus:dev**

5.2. En un navegador web, por ejemplo Firefox, abre el http://localhost:8888 y valida que el Keycloak esta ejecutandose. Esto es necesario para que la aplicación front-end haga redirect a los usuarios a la página de login de Keycloak.

5.3. Inicia la aplicación front end con: **npm install** y luego **npm run dev**. Esto levantará la aplicación en el puerto 9000 y abre el http://localhost:9000. Usa el usuario user y password redhat. Veras el dashboard con 4 speakers.

5.4. Clic en Add un speaker. Ingresa tus nombres y apellidos en first name y last name, luego click en Confirm. Se te mostrará un error, porque el usuario user no esta autorizado a crear speakers. Cierra todos los navegadores para deslogueearte.

5.5. En una nueva ventana abre http://localhost:9000. Usa el usuario superuser y password redhat.

5.6. Clic en Add a speaker. Ingresa tus nombres y apellidos en firstname y lastname respectivamente. Luego haz click en Confirm. La llamada funcionará exitosamente, porque el usuario superuser si tiene permiso para crear spekers. Cierra la ventana del navegador.

5.7. Retorna a la terminal de windows que ejecuta el servicio speaker y luego presiona la letra q para terminar la aplicación.

Si lograste llegar aquí. Felicitaciones haz terminado tu security review.

José