

Estàndards

Desenvolupament d'aplicacions del GOIB

Guia de configuració de l'entorn de desenvolupament



G CONSELLERIA
O ADMINISTRACIONS
I PÚBLIQUES I
B MODERNITZACIÓ
/ DIRECCIÓ GENERAL
MODERNITZACIÓ I
ADMINISTRACIÓ DIGITAL

Palma, maig de 2020

Índice

HISTORIAL DE VERSIONS.....	3
1. INTRODUCCIÓ.....	4
2. OPENJDK 11.....	4
2.1. Instal·lació.....	4
3. JBOSS EAP 7.2.....	5
3.1. Instal·lació.....	5
3.2. Configuració de datasources.....	5
3.3. Canvis importants respecte a la versió EAP 5.2.....	8
4. KEYCLOAK 6.0.1.....	9
4.1. Instal·lació manual.....	9
4.2. Imatge docker.....	10
4.3. Exemple de configuració.....	11
5. CONNEXIÓ JBOSS AMB KEYCLOAK.....	16
5.1. Exemple de configuració del connector.....	17
6. RESOLUCIÓ D'ERRORS.....	21
6.1. Context de només lectura.....	21
6.2. Error d'autenticació.....	21
6.3. Activació de serveis Java EE addicionals.....	21

Historial de versions

Data	Versió	Descripció	Autor
15/01/20	9.0	Primera versió	DGMAD
28/02/20	9.1	Correcció d'errors als exemples Nova imatge docker de keycloak	DGMAD
28/05/20	9.2	Afegir informació de proxy per imatges Docker	DGMAD

1. Introducció

La finalitat d'aquest document és proporcionar una guia de configuració de l'entorn tecnològic per fer servir els estàndards de desenvolupament del Govern de les Illes Balears (GOIB). És una ajuda per al desenvolupador, el seu ús no és obligatori.

Les novetats més rellevants dels estàndards de desenvolupament són:

- **OpenJDK 11** com a plataforma de desenvolupament (enlloc de Java SE 7).
- **JBoss EAP 7.2** com a servidor d'aplicacions (enlloc de JBoss EAP 5.2).
- **Keycloak 6.0.1** com a sistema d'administració d'identitats i accés (enlloc del sistema Seycon propi).

2. OpenJDK 11

OpenJDK 11 és la versió lliure de la plataforma de desenvolupament Java SE Development Kit 11.

2.1. Instal·lació

1. Accedir a l'adreça <https://jdk.java.net/java-se-ri/11> i escollir entre versió Linux/x64 o Windows/x64 (a aquest manual farem servir la versió Windows).
2. Descarregar el fitxer **openjdk-11+28_windows-x64_bin.zip**.
3. Extreure el fitxer al directori **C:\Program Files\Java**.
4. Establir la variable d'entorn **JAVA_HOME** amb el valor **C:\Program Files\Java\jdk-11** (això és necessari ja que tots els scripts de JBoss fan referència a aquesta variable).
5. Afegir el valor **%JAVA_HOME%\bin** a la variable d'entorn **PATH**.

Nota: Aquesta és una implementació de referència de OpenJDK 11. Les versions actualitzades es poden descarregar des de la web d'Oracle (requereix llicència) o bé d'altres entitats que mantenen builds actualitzats de OpenJDK 11 (com per exemple les mantingudes per AdoptOpenJDK).

3. JBoss EAP 7.2

Red Hat JBoss Enterprise Application Platform 7.2 (JBoss EAP 7.2) és una implementació certificada de les especificacions completes i del perfil web de Java Enterprise Edition 7 (Java EE 7).

3.1. Instal·lació

1. Accedir a l'adreça <https://developers.redhat.com/products/eap/download/>.
2. Descarregar el fitxer **jboss-eap-7.2.0-installer.jar** (és necessari registrar-se a la pàgina de RedHat amb un compte gratuït).
3. Executar l'assistent d'instal·lació.
4. Especificar el directori d'instal·lació del JBoss (per exemple: **C:\Desarrollo\jboss-eap-7.2**).
5. Donar d'alta l'usuari administrador del JBoss (per exemple: **admin**). Alternativament, aquest usuari es pot crear amb l'script **JBOSS_HOME\bin\add-user**.
6. Establir la variable d'entorn **JBOSS_HOME** amb el valor del directori d'instal·lació del JBoss (per exemple: **C:\Desarrollo\jboss-eap-7.2**).

Nota: Les releases notes amb els bugs solucionats a la versió 7.2 són a l'adreça https://access.redhat.com/documentation/en-us/red_hat_jboss_enterprise_application_platform/7.2/. Per evitar errors, es recomana aplicar el darrer patch publicat.

3.2. Configuració de datasources

A continuació es descriu el procés de configuració de datasources per sistemes gestors de base de dades Oracle i PostgreSQL segons els estàndards de base de dades del GOIB.

Oracle

1. Crear el directori **JBOSS_HOME\modules\system\layers\base\com\oracle\main**.
2. Descarregar el fitxer **ojdbc8.jar** al directori anterior des de <https://www.oracle.com/technetwork/database/features/jdbc/jdbc-ucp-122-3110062.html> (s'han d'acceptar els termes i condicions i tenir un compte en el web d'Oracle).
3. Crear el fitxer **module.xml** al directori anterior amb el següent contingut:

```
<module xmlns="urn:jboss:module:1.0" name="com.oracle">
  <resources>
    <resource-root path="ojdbc8.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

PostgreSQL:

1. Crear el directori **JBOSS_HOME\modules\system\layers\base\org\postgresql\main**.
2. Descarregar el fixter **postgresql-42.2.5.jar** al directori anterior des de <https://jdbc.postgresql.org/download.html> (es recomana la descàrrega de la versió 42.2.5 que es troba en la taula de Other versions en la columna JDBC 4.2).
3. Crear el fitxer **module.xml** al directori anterior amb el següent contingut:

```
<module xmlns="urn:jboss:module:1.0" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-42.2.5.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Oracle y PostgreSQL:

1. Afegir la següent configuració de «drivers» al fitxer **JBOSS_HOME\standalone\configuration\standalone.xml**.

```
<datasources>
  ...
  <drivers>
    <driver name="h2" module="com.h2database.h2">
      <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
    </driver>
    <!-- GOIB drivers -->
    <driver name="oracle" module="com.oracle">
      <xa-datasource-class> oracle.jdbc.xa.client.OracleXADataSource
    </xa-datasource-class>
    </driver>
```

```
<driver name="postgresql" module="org.postgresql">
  <xa-datasource-class>org.postgresql.xa.PGXADatasource
</xa-datasource-class>
</driver>
</drivers>
...
</datasources>
```

2. Reiniciar el JBoss (si es trobàs en marxa) i afegir els datasources que facin falta. A aquest punt tenim tres **opcions**:

2.1. **Opció recomanada** (segons els estàndards de base de dades): crear un fitxer XML independent (anomenat **nomAplicacio-ds.xml**) al directori **JBOSS_HOME\standalone\deployments** per a cada aplicació i afegir-hi el següent contingut:

Per un datasource de tipus **Oracle** el contingut seria el següent:

```
<datasource jndi-name="java:jboss/datasources/codiAppDS" pool-name="codiAppDS"
enabled="true" use-java-context="true">
  <connection-url>jdbc:oracle:thin://host:1523/nombd</connection-url>
  <driver>oracle</driver>
  <security>
    <user-name>userapp</user-name>
    <password>pass</password>
  </security>

  <new-connection-sql>
  BEGIN
  EXECUTE IMMEDIATE 'ALTER SESSION SET CURRENT_SCHEMA = NOMBD';
  END;
  </new-connection-sql>
</datasource>
```

Per un datasource de tipus **PostgreSQL** el contingut seria el següent:

```
<datasource jndi-name="java:jboss/datasources/codiAppDS" pool-name="codiAppDS"
enabled="true" use-java-context="true">
  <connection-url>jdbc:postgresql://host:5432/nombd</connection-url>
  <driver>postgresql</driver>
  <security>
    <user-name>userapp</user-name>
    <password>pass</password>
  </security>

  <new-connection-sql>
```

```
BEGIN  
EXECUTE IMMEDIATE 'ALTER SESSION SET CURRENT_SCHEMA = NOMBD';  
END;  
</new-connection-sql>  
</datasource>
```

2.2. Afegir els datasources anteriors dins l'etiqueta <datasources> del fitxer **JBOSS_HOME\standalone\configuration\standalone.xml**.

2.3. Fer servir la **consola d'administració** del JBoss (<http://localhost:9990/console/index.html>). Cal tenir en compte que els valors per defecte que inclou no són exactament els mateixos.

3.3. *Canvis importants respecte a la versió EAP 5.2*

1. Per iniciar el JBoss s'ha d'executar l'script **JBOSS_HOME\bin\standalone.bat** a entorns Windows o l'script **JBOSS_HOME\bin\standalone.sh** a entorns Unix/Linux.
2. Per desplegar aplicacions s'ha de copiar el fitxer EAR dins del directori **JBOSS_HOME\standalone\deployments**.
3. El fitxer de configuració principal es troba a **JBOSS_HOME\standalone\configuration\standalone.xml**.

4. Keycloak 6.0.1

Keycloak és un producte de programari de codi obert que permet l'inici de sessió únic (IdP) amb Identity Management i Access Management. Al GOIB farem servir, per un costat, Keycloak com a servidor esperant peticions d'autenticació, i per altre, JBoss 7.2 EAP amb un adaptador per poder connectar-lo amb Keycloak.

Nota: El servidor es pot instal·lar i configurar de manera manual o, alternativament, fer servir una imatge **docker** adaptada a l'entorn de desenvolupament del GOIB (veure secció 4.2).

4.1. Instal·lació manual

1. Accedir a l'adreça <https://www.keycloak.org/downloads.html>.
2. Descarregar el **Standalone Server Distribution versió 6.0.1**.
3. Extreure el fitxer **keycloak-6.0.1.zip** al directori d'instal·lació (per exemple: **C:\Desarrollo\keycloak-6.0.1**).
4. Establir la variable d'entorn **KEYCLOAK_HOME** amb el valor del directori d'instal·lació (per exemple: **C:\Desarrollo\keycloak-6.0.1**).
5. Keycloak és un JBoss modificat. Perquè no hi hagi conflictes de ports entre el JBoss EAP 7.2 i el JBoss del Keycloak, a un dels dos servidors (a aquest manual farem el canvi al Keycloak) s'ha de substituir el valor de la propietat **port-offset**. Això es degut a que JBoss EAP 7.2 i Keycloak fan servir els mateixos jocs de ports, en concret:
 - 8080/8443 per accés HTTP/HTTPS
 - 9990/9993 per configuració HTTP/HTTPS
 - 8009 per AJP

Amb aquest canvi tots els valors dels ports del servidor sumarien 100 al seu valor original:

- 8180/8543 per accés HTTP/HTTPS
- 10090/10093 per configuració HTTP/HTTPS
- 8109 per AJP

El canvi dels ports es pot fer de dues maneres :

- a) Modificant el paràmetre port-offset de la propietat **socket-binding-group** al fitxer **KEYCLOAK_HOME\standalone\configuration\standalone.xml**

```
<socket-binding-group name="standard-sockets" default-interface="public" port-offset="{jboss.socket.binding.port-offset:0}">
```

```
<socket-binding-group name="standard-sockets" default-  
interface="public" port-offset="{jboss.socket.binding.port-  
offset:100}">
```

b) Iniciant el servidor amb la comanda d'execució **KEYCLOAK_HOME\bin\standalone.bat -Djboss.socket.binding.port-offset=100**.

6. Reemplaçar el nom de la variable JBOSS_HOME per KEYCLOAK_HOME a l'script **KEYCLOAK_HOME\bin\standalone.bat**.
7. Iniciar el Keycloak executant l'script **KEYCLOAK_HOME\bin\standalone.bat**.
8. Accedir a la consola d'administració del Keycloak (per defecte amb els ports desplaçats: <http://localhost:8180/auth>).
9. Si no tenim cap usuari administrador, cal afegir-lo per primera vegada mitjançant la pròpia consola d'administració o mitjançant l'script **KEYCLOAK_HOME\bin\add-user-keycloak**.

4.2. Imatge docker

Alternativament, el GOIB proveeix una imatge docker que conté un servidor de Keycloak ja configurat amb el realm, clients, usuaris, i rols de l'exemple descrit a l'apartat 4.3.

Si tenim l'entorn de docker instal·lat (<https://docs.docker.com/get-docker/>), podem crear i executar un contenidor d'aquesta imatge executant la següent instrucció:

```
$ docker run -p 8180:8180 goib/keycloak:6.0.1
```

L'usuari administrador per defecte és **admin** amb contrasenya **admin** (es pot modificar afegint els paràmetres `-e KEYCLOAK_USER=user` i `-e KEYCLOAK_PASSWORD=password`).

Nota: Per tal de poder descarregar les imatges docker dins del GOIB cal configurar el servidor intermediari (proxy).

En cas d'un sistema operatiu **Windows** el procés seria el següent:

1. Obrim una finestra Powershell des de «Inici->Executar->PowerShell»
2. Executem la següent instrucció:

```
[Environment]::SetEnvironmentVariable("HTTP_PROXY",  
"http://username:password@rproxy1.caib.es:3128/", [EnvironmentVariableTarget]::Machine)
```

3. Reiniciem el servei

```
$ Restart-Service docker
```

En cas d'un sistema operatiu **Unix/Linux**:

1. Creem el fitxer `/etc/systemd/system/docker.service.d/http-proxy.conf` amb el següent contingut:

```
[Service]
Environment="HTTP_PROXY=http://username:password@rproxy1.caib.es:3128/"
```

2. Carregam la configuració:

```
$ sudo systemctl daemon-reload
```

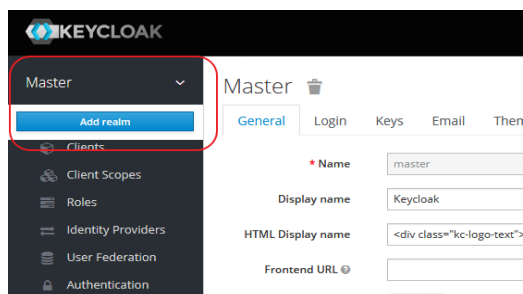
3. Reiniciam el servei:

```
$ sudo systemctl restart docker
```

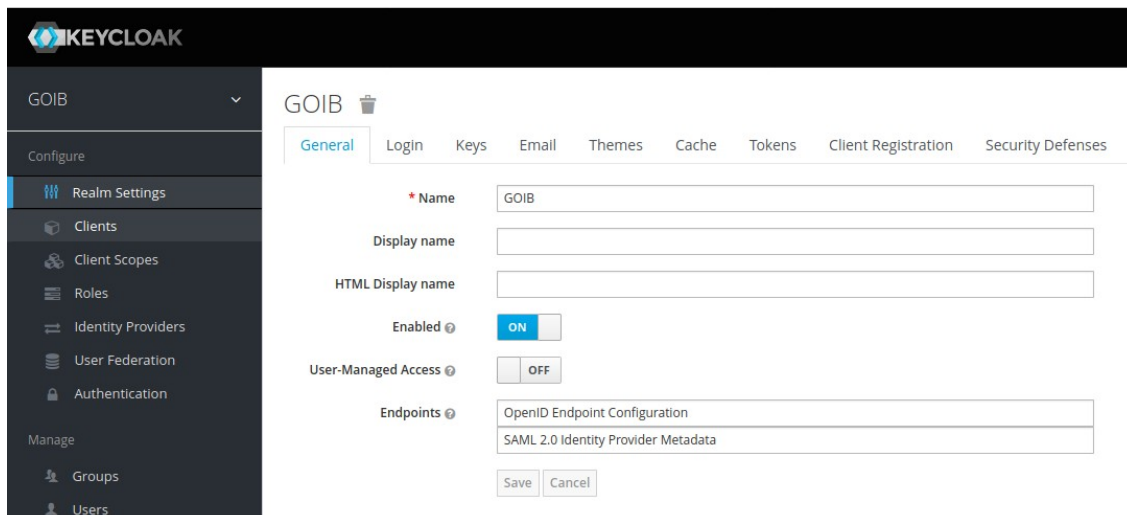
4.3. Exemple de configuració

A continuació es mostra un exemple de configuració per controlar l'accés a una aplicació denominada **goibusuari** que mostra informació de l'usuari autenticat i proporciona diversos mètodes amb una API REST. A l'exemple crearem un domini d'actuació (realm) i dos clients (un per al backoffice i un per l'API REST).

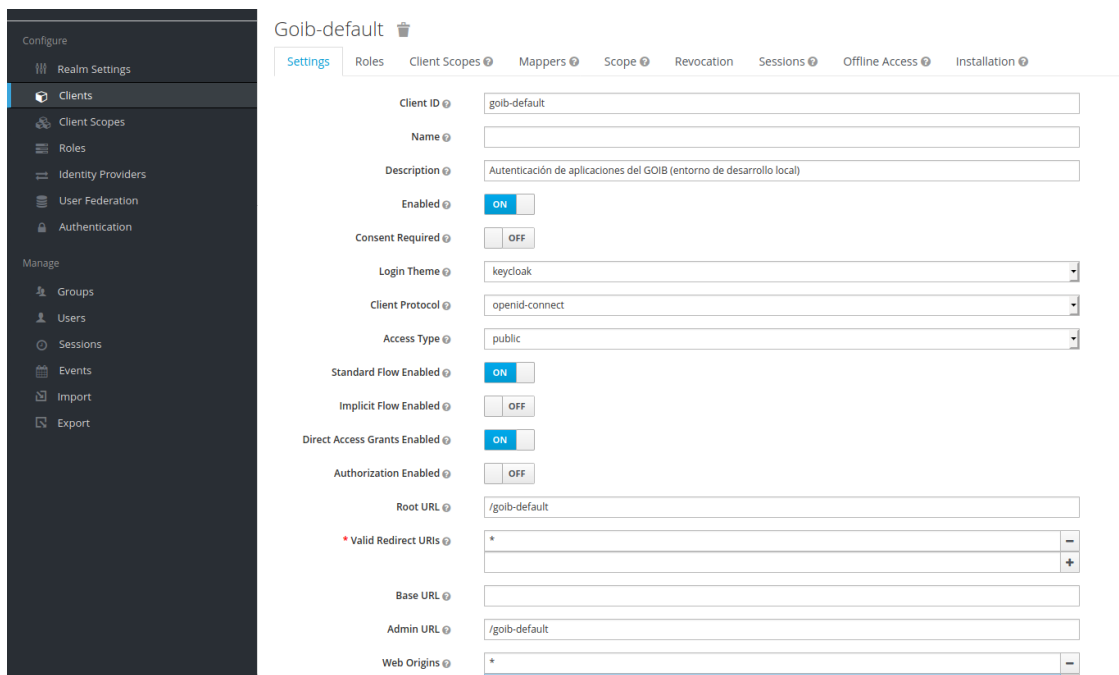
1. Accedim a la consola d'administració <http://localhost:8180/auth>.
2. Pitjam sobre el desplegable del menú i seleccionam «**Add realm**»



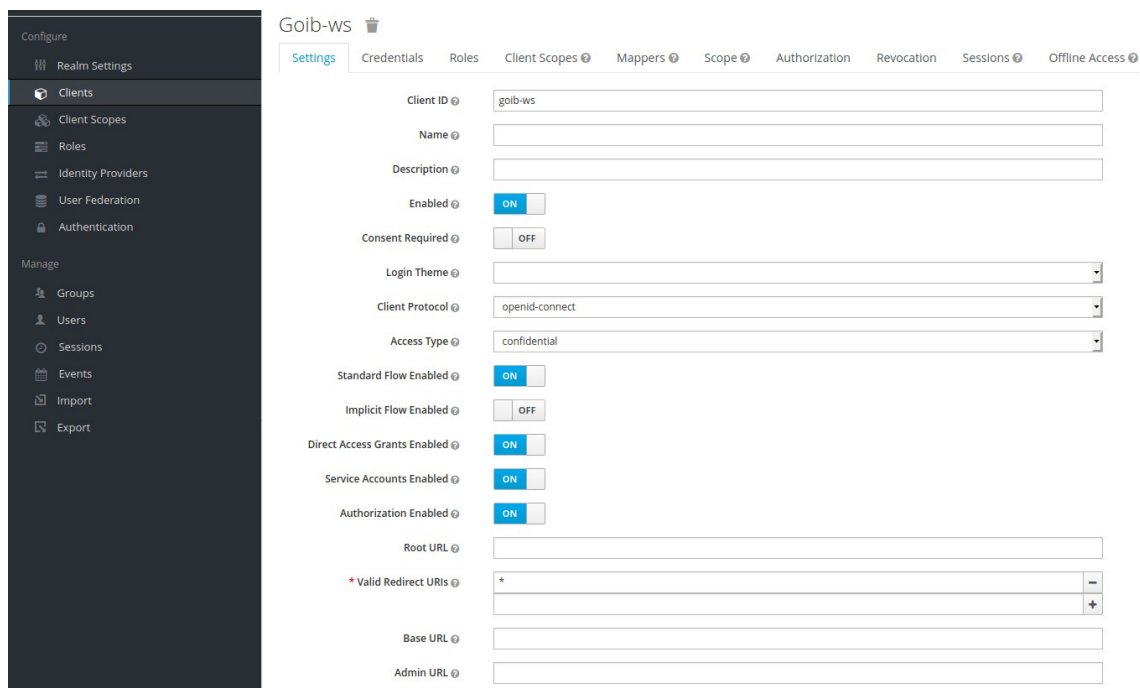
3. Li posam de nom **GOIB** i deixam la resta de camps amb els valors per defecte.



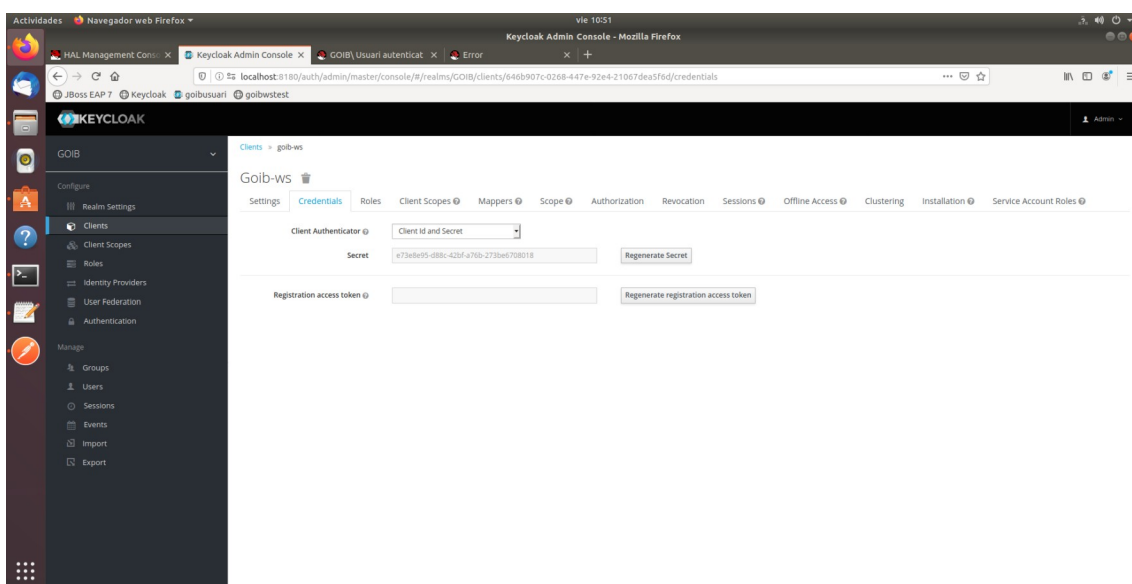
- Afegim el client **goib-default** per al backoffice amb els següents valors: **/goib-default** al camp **ROOT URL**, i ***** al camp **Valid Redirect URIs**; la resta de camps deixam el valor per defecte.



- Afegim el client **goib-ws** per als serveis REST amb els següents valors: **confidential** al camp **Access Type**, ***** al camp **Valid Redirect URIs**, i les opcions **Authorization Enabled** i **Service Accounts Enabled** activades; la resta de camps deixam el valor per defecte.

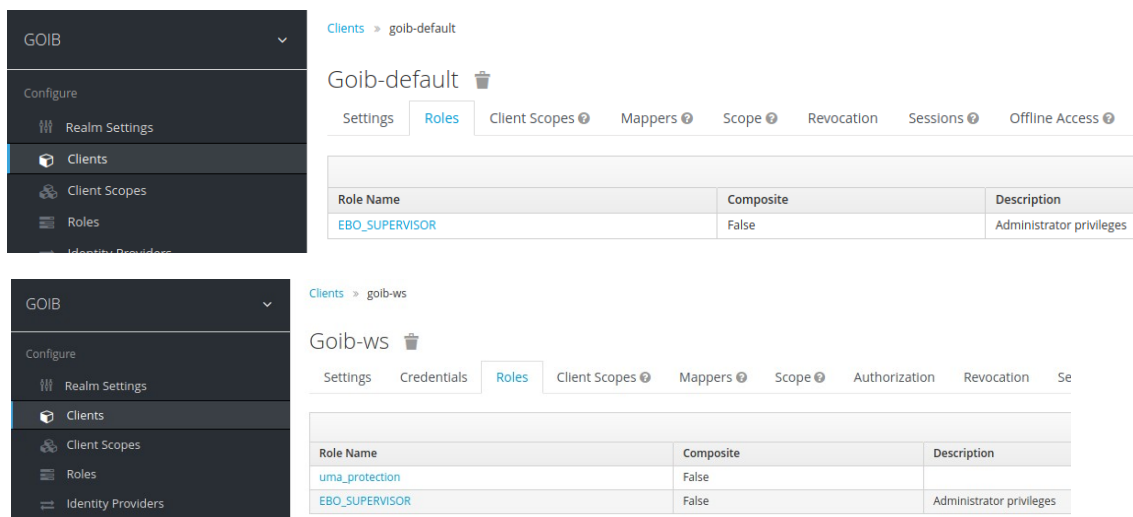


6. Copiam la contrasenya («credentials») generada automàticament ja que la necessitarem més endavant per configurar el JBoss.



7. Afegim el rol **EBO_SUPERVISOR** dins de cada client (podem configurar rols a nivell de realm perquè els usuaris tinguin accés a tots els mòduls o a nivell de client perquè els usuaris tinguin accés només a un mòdul en particular).

Nota: Al capítol 5 veurem com configurar el connector de JBoss amb Keycloak per establir el nivell d'autenticació fent servir el paràmetre «use-resource-role-mappings».



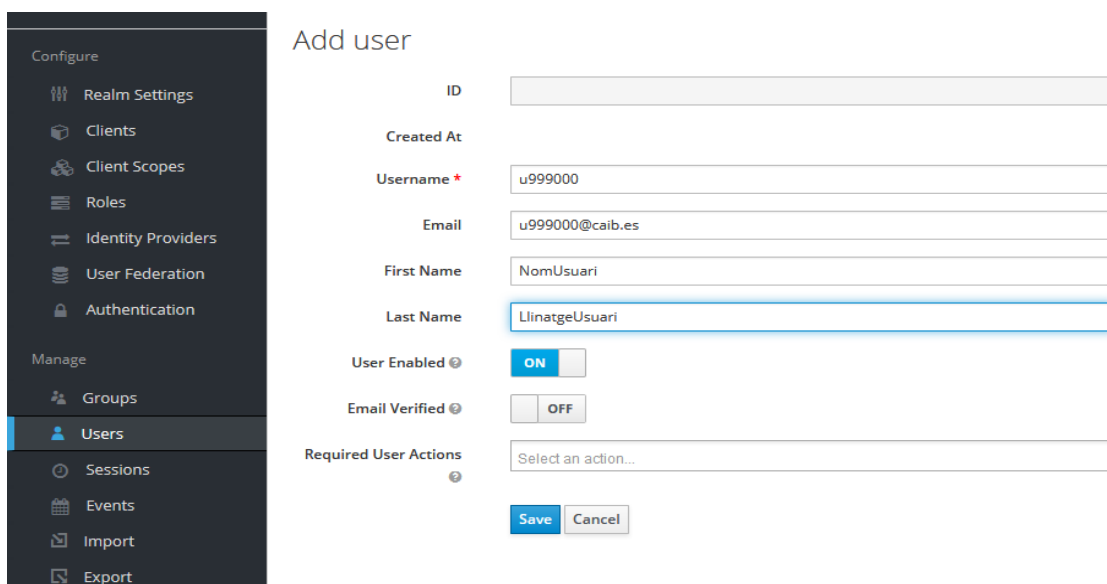
goib-default Roles

Role Name	Composite	Description
EBO_SUPERVISOR	False	Administrator privileges

goib-ws Roles

Role Name	Composite	Description
uma_protection	False	
EBO_SUPERVISOR	False	Administrator privileges

8. Afegim l'usuari **u999000** i li assignam la mateixa contrasenya dins l'apartat «credentials». **Si no li assignam contrasenya no es podrà autenticar.** Aquest usuari el farem servir per autenticar-nos al backoffice.



Add user

ID:

Created At:

Username *

Email

First Name

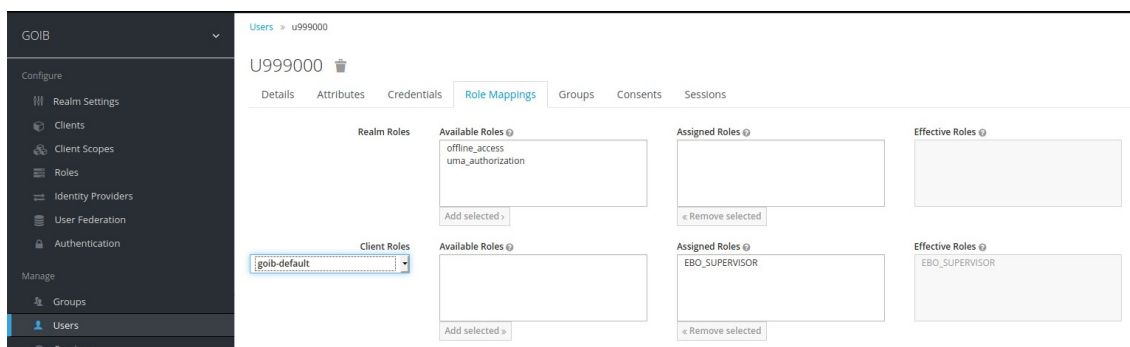
Last Name

User Enabled

Email Verified

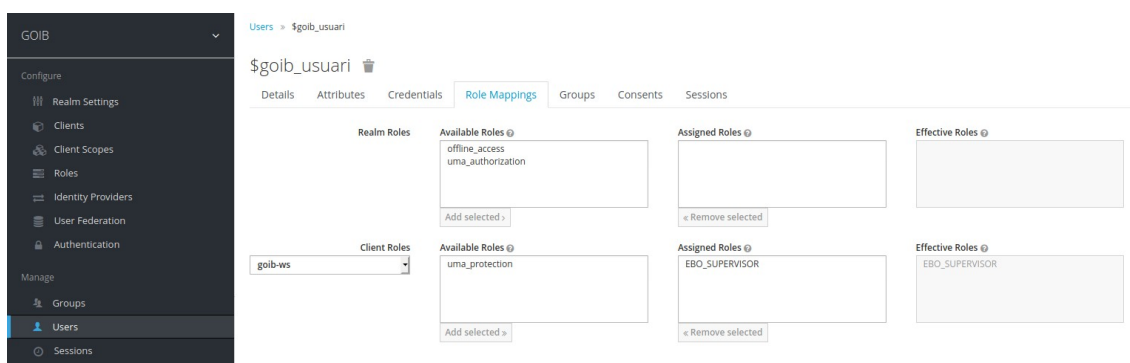
Required User Actions

9. Assignam, a l'usuari **u999000**, el rol **EBO_SUPERVISOR** per al client **goib-default**.



10. Afegim l'usuari **\$goib_usuari** i li assignam la mateixa contrasenya dins l'apartat «**credentials**». Si no li assignam contrasenya no es podrà autenticar. Aquest usuari el farem servir per autenticar els serveis REST.

11. Assignam, a l'usuari **\$goib_usuari**, el rol **EBO_SUPERVISOR** per al client **goib-WS**.



5. Connexió JBoss amb Keycloak

Per connectar JBoss EAP 7.2 amb Keycloak s'ha d'instal·lar un adaptador.

1. Accedir a l'adreça <https://www.keycloak.org/downloads.html>.
2. Descarregar el **Client Adapter** de Keycloak (OPENID CONNECT) per a JBoss 7 EAP.
3. Extreure el fitxer **keycloak-wildfly-adapter-dist-6.0.1.zip** al `JBOSS_HOME`. Al directori `JBOSS_HOME\bin` s'afegiran els següents executables:
 - `adapter-install-offline.cli`
 - `adapter-install.cli`
 - `adapter-elytron-install-offline.cli`
 - `adapter-elytron-install.cli`

Important: Actualment, la versió amb ELYTRON té UN BUG i dona problemes amb els EJBs. Per tant, es desaconsella fer-lo servir. La diferència entre les versions «normal» i les «offline» és que el seu èxit depèn de si el JBoss està en marxa o no, respectivament.

4. Amb el JBoss aturat, executar la comanda **`jboss-cli.bat -file=adapter-install-offline.cli`**.

```
C:\DesarrolloSimo\jboss-eap-7.2\bin>jboss-cli.bat --file=adapter-install-offline.cli
OpenJDK 64-Bit Server VM warning: Ignoring option PermSize; support was removed in 8.0
OpenJDK 64-Bit Server VM warning: Ignoring option MaxPermSize; support was removed in 8.0
{"outcome" => "success"}
{"outcome" => "success"}
{"outcome" => "success"}
{"outcome" => "success"}
{"outcome" => "success"}
Presione una tecla para continuar . . .
```

5. Per últim, s'ha d'afegir la següent configuració dins el «subsystem» que fa referència al keycloak al fitxer `JBOSS_HOME\standalone\configuration\standalone.xml`:

```
<subsystem xmlns="urn:jboss:domain:keycloak:1.1">
  <realm name="NOM_REALM">
    <auth-server-url>URL_KEYCLOAK</auth-server-url>
    <ssl-required>NONE/EXTERNAL/ALL</ssl-required>
  </realm>
  <secure-deployment name="NOM_WAR_BACKOFFICE.war">
    <realm>NOM_REALM</realm>
    <resource>NOM_CLIENT_1</resource>
    <use-resource-role-mappings>TRUE/FALSE</use-resource-role-mappings>
    <public-client>true</public-client>
    <verify-token-audience>true</verify-token-audience>
  </secure-deployment>
```



```
<secure-deployment name="NOM_WAR_API_REST.war">
  <realm>NOM_REALM</realm>
  <resource>NOM_CLIENT_2</resource>
  <use-resource-role-mappings>TRUE/FALSE</use-resource-role-mappings>
  <bearer-only>>true</bearer-only>
  <enable-basic-auth>true</enable-basic-auth>
  <principal-attribute>preferred_username</principal-attribute>
  <credential name="secret">password</credential>
</secure-deployment>
</subsystem>
```

Els valors a configurar són els següents:

- **realm name:** nom del REALM (domini d'actuació del keycloak).
- **uth-server-url:** Url del servidor Keycloak (si es té en local, <http://localhost:8181/auth>).
- **ssl-required:** Els valors possibles són:
 - NONE: No es requereix HTTPS per cap adreça IP de client.
 - EXTERNAL: Les adreces IP privades i de localhost poden accedir sense HTTPS.
 - ALL: Es requereix HTTPS per totes les adreces IP.
- **secure-deployment:** configuració d'un model identificat pel nom del WAR. S'hi ha d'especificar el nom de realm sota el qual fa feina el mòdul.
- **resource:** Nom de CLIENT o mòdul a que es fa referència dins el Keycloak.
- **use-resource-role-mappings:**
 - TRUE: avalua el rol a nivell de CLIENT.
 - FALSE: avalua el rol a nivell de REALM.

Per al cas d'un war de tipus **backoffice** s'han d'afegir les propietats **public-client** i **verify-token-audience** amb valor **true**.

Per al cas d'un war de tipus **API REST** s'han d'afegir les propietats **bearer-only**, **enable-basic-auth**, i establir el **credential "secret"** amb el valor de la contrasenya del client API REST.

5.1. Exemple de configuració del connector.

A continuació es mostra un exemple per connectar l'aplicació **goibusuari** amb el servidor Keycloak local descrit a l'apartat «4.2. Exemple de configuració»; es a dir, farem servir el realm **GOIB** i els clients **goib-default** i **goib-ws**.

1. Suposant que volem accedir només a nivell de client, la configuració del fitxer **JBOSS_HOME\standalone\configuration\standalone.xml** seria la següent:

```
<subsystem xmlns="urn:jboss:domain:keycloak:1.1">
  <realm name="GOIB">
    <auth-server-url>http://localhost:8180/auth</auth-server-url>
    <ssl-required>EXTERNAL</ssl-required>
  </realm>

  <secure-deployment name="userinfo.war">
    <realm>GOIB</realm>
    <resource>goib-default</resource>
    <use-resource-role-mappings>true</use-resource-role-mappings>
    <public-client>true</public-client>
    <verify-token-audience>true</verify-token-audience>
    <principal-attribute>preferred_username</principal-attribute>
  </secure-deployment>

  <secure-deployment name="rest.war">
    <realm>GOIB</realm>
    <resource>goib-ws</resource>
    <use-resource-role-mappings>true</use-resource-role-mappings>
    <bearer-only>true</bearer-only>
    <enable-basic-auth>true</enable-basic-auth>
    <principal-attribute>preferred_username</principal-attribute>
    <credential name="secret">e73e8e95-d88c-42bf-a76b-273be6708018</credential>
  </secure-deployment>
</subsystem>
```

2. L'aplicació **goibusuari** ja té configurat el rol **EBO_SUPERVISOR** dins dels paquets **userinfo.war** i **rest.war**; en concret, dins del fitxer **src\main\webapp\WEB-INF\web.xml**:

```
<security-constraint>

  <web-resource-collection>
    <web-resource-name>UserInfo</web-resource-name>
    <url-pattern>/*</url-pattern>
    <http-method>POST</http-method>
    <http-method>GET</http-method>
  </web-resource-collection>

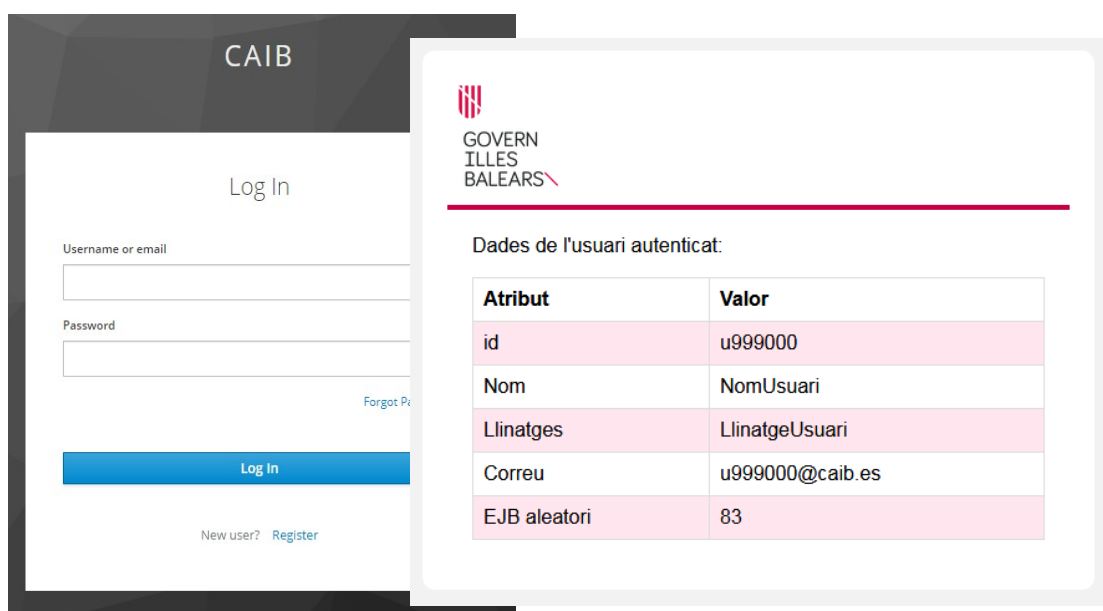
  <auth-constraint>
    <role-name>EBO_SUPERVISOR</role-name>
  </auth-constraint>
</security-constraint>

<login-config>
```

```
<auth-method>KEYCLOAK</auth-method>
<realm-name>Autenticacio</realm-name>
</login-config>

<security-role>
  <role-name>EBO_SUPERVISOR</role-name>
</security-role>
```

- Desplegar el fitxer **goibusuari.ear**¹ dins del directori **JBoss_HOME\standalone\deployments** del JBoss EAP 7.2.
- Per provar l'accés autenticat al backoffice, accedim a l'adreça <http://localhost:8080/goibusuari/>. Apareixerà una finestra on s'han d'inserir les credencials de l'usuari (en aquest cas, l'usuari **u999000** creat a la secció 4.2). Si les dades són correctes, apareixerà una finestra amb les dades de l'usuari autenticat.



- Per provar l'accés autenticat a l'API REST, hem de fer servir un client (com SoapUI) que permeti fer cridades als mètodes proporcionats. A la secció «Authorization» de la cridata hem d'establir l'usuari **\$goib-usuari** amb la seva contrasenya.

Els mètodes proporcionats per l'API són:

- POST: <http://localhost:8080/goibwstest/rest/persona/add>. Afegeix una persona en format JSON, per exemple:

¹Aquest EAR el podeu trobar al directori doc del ProjecteBase <https://github.com/GovernIB/projectebase>.



```
{  
  "id":1,  
  "nom":"Usuari de prova",  
  "any":2020  
}
```

- GET: <http://localhost:8080/goibwstest/rest/persona/{id}/delete>. Elimina la persona amb la indicada.
- GET: <http://localhost:8080/goibwstest/rest/persona/getAll>. Retorna totes les persones afegides prèviament.

6. Resolució d'errors.

6.1. Context de només lectura

Si durant l'inici del JBoss aparegués un error al WeldStartService de «Contexto de només lectura», s'ha de afegir el paràmetre **require-bean-descriptor="true"** al subsistema **Weld** del fitxer `JBOSS_HOME\standalone\configuration\standalone.xml`.

```
<subsystem xmlns="urn:jboss:domain:weld:4.0" require-bean-descriptor="true"/>
```

6.2. Error d'autenticació

L'usuari creat al Keycloak ha de tenir el rol assignat corresponent. A més, cal recordar que si no li assignam contrasenya no es podrà autenticar.

S'ha de verificar que el contingut del fitxer `JBOSS_HOME\standalone\configuration\standalone.xml` sigui correcte (veure exemple a la secció 5.1)

L'aplicació ha de estar configurada correctament (EJBs, fitxers web.xml,...). Per més informació, veure el capítol «3.3 Seguridad de Aplicaciones» del document «Estándares de Desarrollo de aplicaciones del GOIB. Aplicaciones Java EE»

6.3. Activació de serveis Java EE addicionals

Cal tenir en compte que el fitxer `JBOSS_HOME\standalone\configuration\standalone.xml` inicia el JBoss amb una configuració bàsica que no inclou tots els serveis disponibles de JavaEE; per exemple, el més destacable és que no inclou JMS/MDB. La configuració que inclou aquests serveis es troba al fitxer `JBOSS_HOME\standalone\configuration\standalone-full.xml`. Per iniciar amb aquesta configuració cal afegint l'opció "-c" amb el nom de la configuració (per exemple: `JBOSS_HOME\bin\standalone.bat/.sh -c standalone-full.xml`).