

Elytron

Résumé de configuration

Création d'un Realm (Comment on s'authentifie)



Création d'un Domain constitué de 1 ou plusieurs Realm



Création d'un AppDomain constitué d'un domain ou d'une autre méthode d'authentification

Résumé de configuration

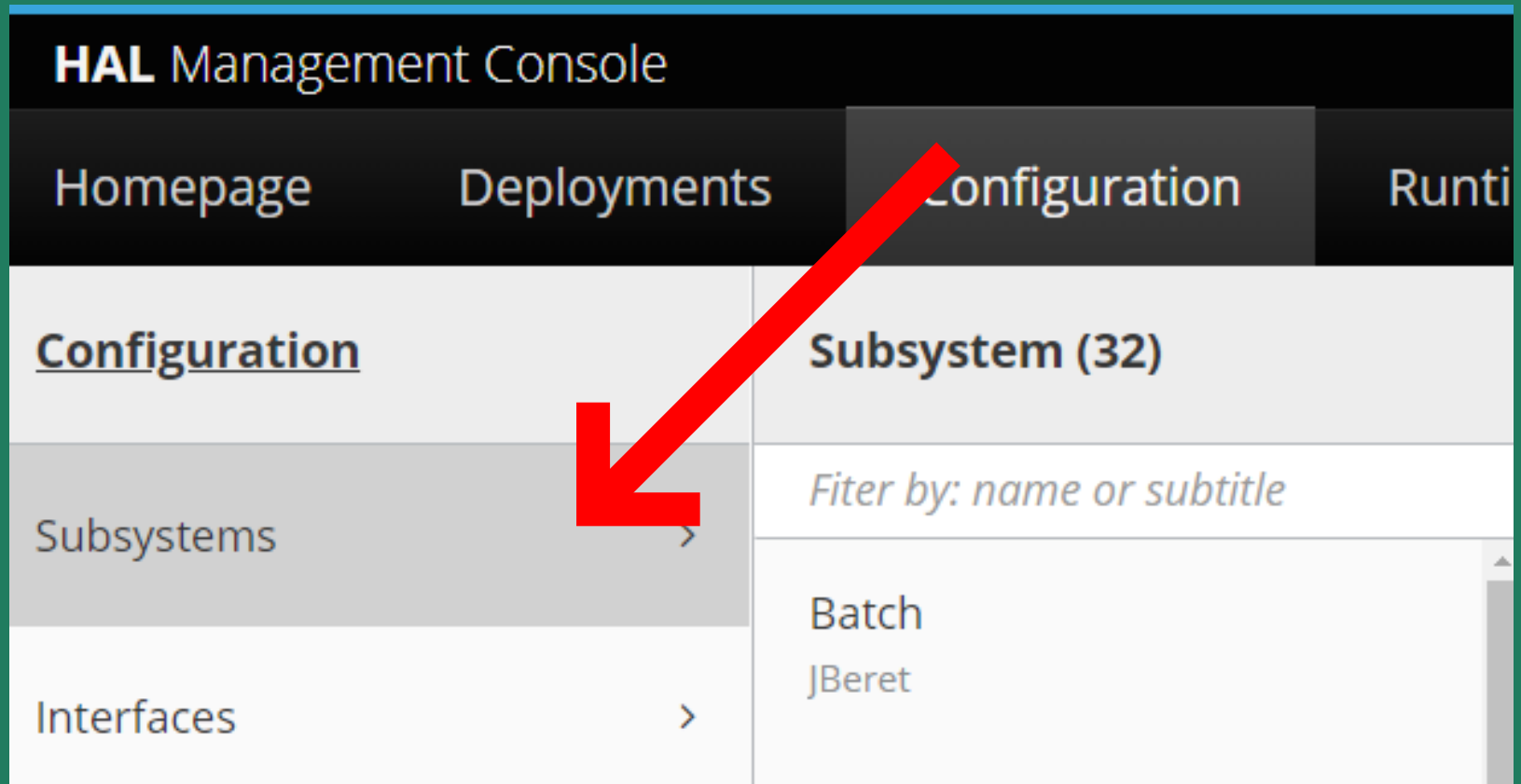
C'est ensuite le nom de l'AppDomain qui sera fourni aux Applications qui souhaitent être sécurisé par Elytron

Exemple 1

Authentification via la base de donnée
en utilisant une connexion JDBC

Création d'un Realm pour JDBC: Le JDBC Realm

Depuis l'interface de management (Port 9990)
Choisissez Configuration -> Subsystem



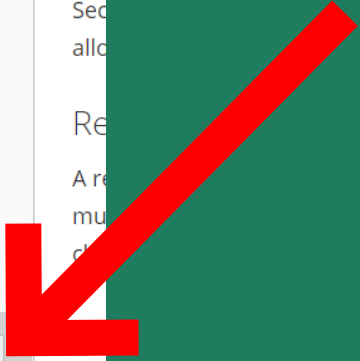
Puis Security

Configuration	<u>Subsystem (32)</u>	Settings
Subsystems >	<i>Fiter by: name or subtitle</i>	<i>Filter</i>
Interfaces >	MicroProfile JWT Smallrye	Global Settings
Socket Bindings >	Microprofile Config Smallrye	Factories / Transformers
Paths	Naming JNDI	Mappers / Decoders
System Properties	Remoting	Other Settings
	Request Controller	Security Realms
	Resource Adapters >	
	Security > Elytron	
	Security Manager	
	Transaction	

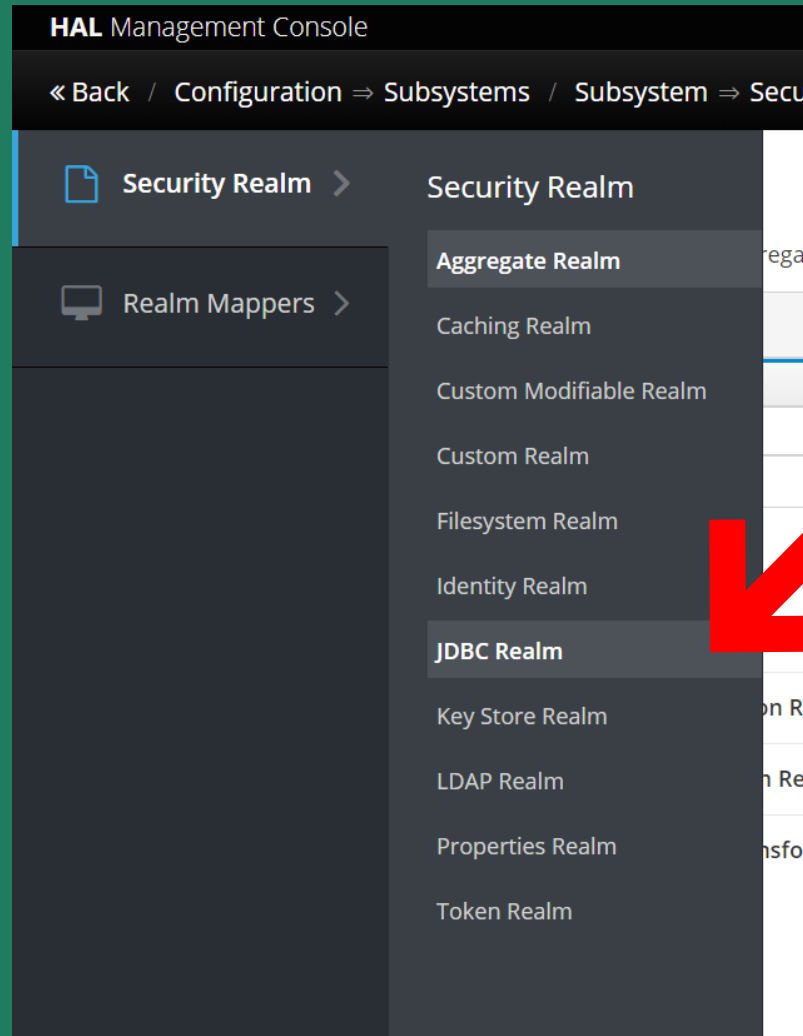
Puis Security Realms -> View

Homepage Deployments **Configuration** Runtime Patching Access Control

Configuration	Subsystem (32)	Settings
Subsystems >	<i>Filter by: name or subtitle</i>	<i>Filter</i>
Interfaces >	MicroProfile JWT Smallrye	Global Settings
Socket Bindings >	Microprofile Config Smallrye	Factories / Transformers
Paths	Naming JNDI	Mappers / Decoders
System Properties	Remoting	Other Settings
	Request Controller	Security Realms View
	Resource Adapters >	
	Security Elytron >	
	Security Manager	



Ici se trouve toutes les options possibles de base dans wildfly: Auth via fichier, ldap, token et JDBC



Nous allons donc créer notre Realm JDBC en lui donnant un nom et une requête sql (principal query) permettant de récupérer le mot de passe par rapport à un login. Cette requête est à modifier par rapport à votre table utilisateur!

Add JDBC Realm [X]

[Help](#)

Name *

Data Source * [v]

SQL *

Required fields are marked with *

Cancel Add

A noter que le datasource ici choisi est celui créé au préalable et qui est utilisé par l'application dans le persistence.xml

Add JDBC Realm [X]

[? Help](#)

Name *

Data Source * ▼

SQL *

Required fields are marked with *

Cancel Add

Nous allons maintenant préciser la requete SQL, notamment en expliquant au Realm la forme que va prendre le mot de passe ainsi récupéré (en clair, md5,sha2,bcrypt etc)

JDBC Realm	
A security realm definition backed by database using JDBC.	
<input type="text"/>	Showing 1 to 2 of 2 Items
Add Remove	
Name ^	Actions
jdbcrealm	Principal Query
testJDBCRealm	Principal Query

En sélectionnant la requête , on fait apparaître les options

JDBC Realm: jdbcrealm > Principal Query

Principal Query

The authentication query used to authenticate users based on specific key types.

Showing 1 to 1 of 1 Items

Add Remove

SQL ^	Data Source
select password From Utilisateur where pseudo =?	PostgresDS

<< < 1 of 1 > >>

[Attributes](#) Clear Password Mapper Bcrypt Mapper Modular Crypt Mapper Salted Simple Digest Mapper Simple Digest Mapper Scram Mapper

[Edit](#) [Help](#)

SQL

Data Source

Attribute Mapping

Je vous renvoi à la doc officielle pour voir quelles formes peuvent prendre les mots de passe:

https://github.com/wildfly/wildfly/blob/main/docs/src/main/asciidoc/_elytron/Passwords.adoc

JDBC Realm: jdbcrealm > Principal Query

Principal Query

The authentication query used to authenticate users based on specific key types.

Showing 1 to 1 of 1 ItemsAdd Remove

SQL ^	Data Source
select password From Utilisateur where pseudo =?	PostgresDS

<< < 1 of 1 > >>

[Attributes](#) [Clear Password Mapper](#) [Bcrypt Mapper](#) [Modular Crypt Mapper](#) [Salted Simple Digest Mapper](#) [Simple Digest Mapper](#) [Scram Mapper](#)

[Edit](#) [Help](#)

SQL

Data Source

Attribute Mapping

Dans ce premier exemple nous allons choisir de stocker les mots de passe en clear text (mauvaise pratique!!), puis dans un deuxième temps nous modifierons la configuration pour passer à une fonction de hashage.

JDBC Realm: jdbcrealm > Principal Query

Principal Query

The authentication query used to authenticate users based on specific key types.

Showing 1 to 1 of 1 ItemsAdd Remove

SQL ^	Data Source
select password From Utilisateur where ps = ?	PostgresDS

« < 1 of 1 > »

[Attributes](#) [Clear Password Mapper](#) [Bcrypt Mapper](#) [Modular Crypt Mapper](#) [Salted Simple Digest Mapper](#) [Simple Digest Mapper](#) [Scram Mapper](#)

[Edit](#) [Help](#)

SQL

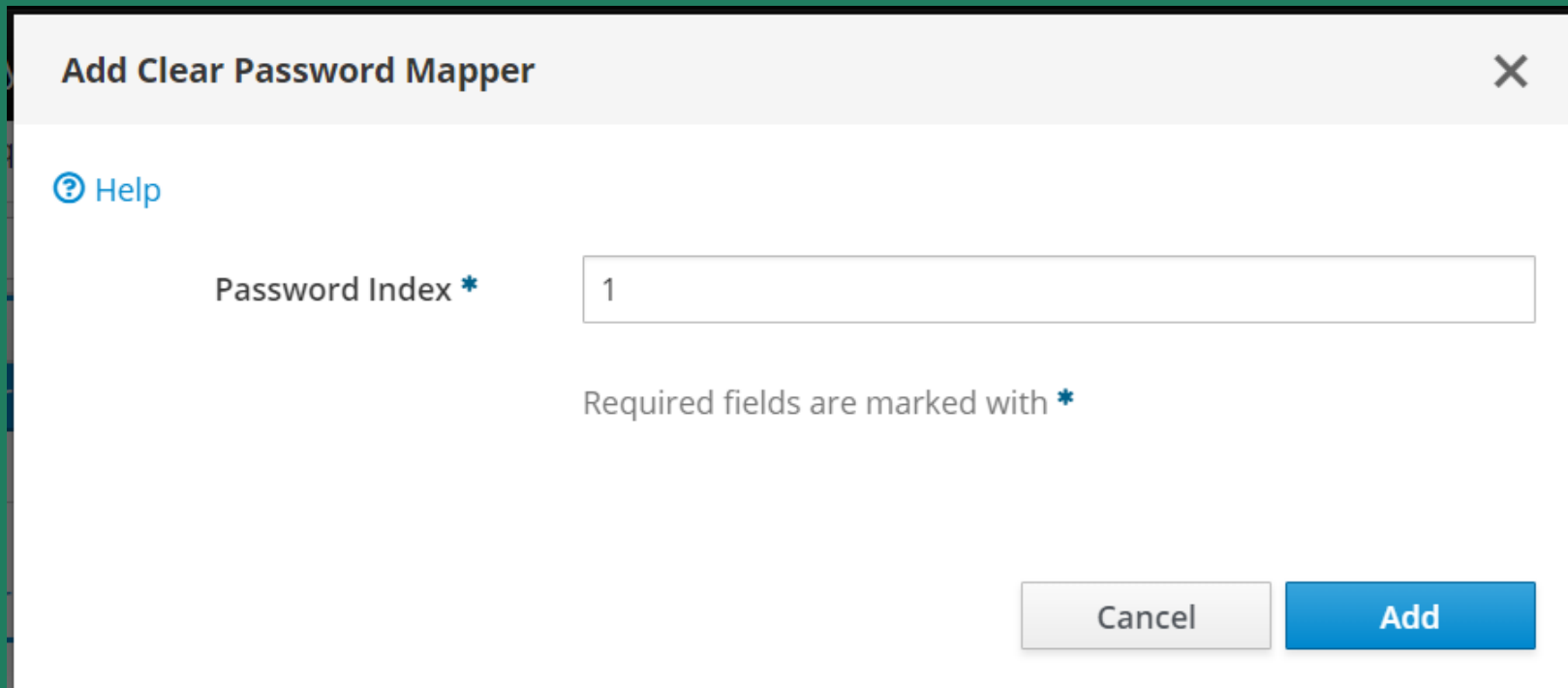
Data Source

Attribute Mapping

Notre principal query est:

```
1 select password from utilisateur where pseudo = ?
```

La colonne password est donc la première de notre requête, on mets 1 dans password Index



Add Clear Password Mapper

[? Help](#)

Password Index *

Required fields are marked with *

Cancel Add

De la même façon nous pouvons spécifier dans la requête où se trouve d'autre élément d'authentification tel que le groupe/role de l'utilisateur.

Pour cela on edite Attribute mapping

Principal Query

The authentication query used to authenticate users based on specific key types.

Showing 1 to 1 of 1 Items

SQL ^

select password From Utilisateur where pseudo =?

Attributes Clear Password Mapper Bcrypt Mapper Modular Crypt Mapper Salted Simple Digest Mapper Simple

? Help

SQL *

select password,'Admin' From Utilisateur where pseudo =?

Data Source *

PostgresDS

Attribute Mapping

groups=2

Add new mappings as to=index pairs. Press ↵ to add and ⌫ to remove them.



On précise que le groups se trouve à l'index 2 de notre requête,
Ici nous avons mis une constant 'Admin', mais on pourrait y
retrouver une colonne classique , ou même une colonne
résultant d'une jointure vers une autre table.

Principal Query

The authentication query used to authenticate users based on specific key types.

Showing 1 to 1 of 1 Items

SQL ^

select password From Utilisateur where pseudo =?

Attributes

Clear Password Mapper

Bcrypt Mapper

Modular Crypt Mapper

Salted Simple Digest Mapper

Simple

? Help

SQL *

select password,'Admin' From Utilisateur where pseudo =?


Data Source *

PostgresDS

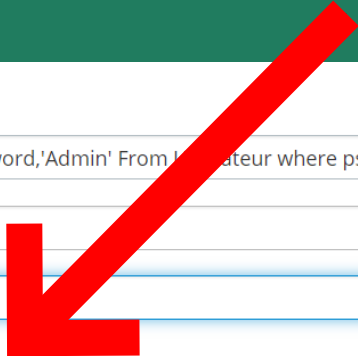
Attribute Mapping

groups=2

Add new mappings as to=index pairs. Press ↵ to add and ✖ to remove them.



Attribute mapping permet d'entrer plusieurs mappings, il faut donc valider chaque couple clé=valeur avec Entrée avant de sauvegarder.



SQL *

Data Source *

Attribute Mapping

groups=2 x

Add new mappings as to=index pairs. Press , to add and ✕ to remove them.

Required fields are marked with *

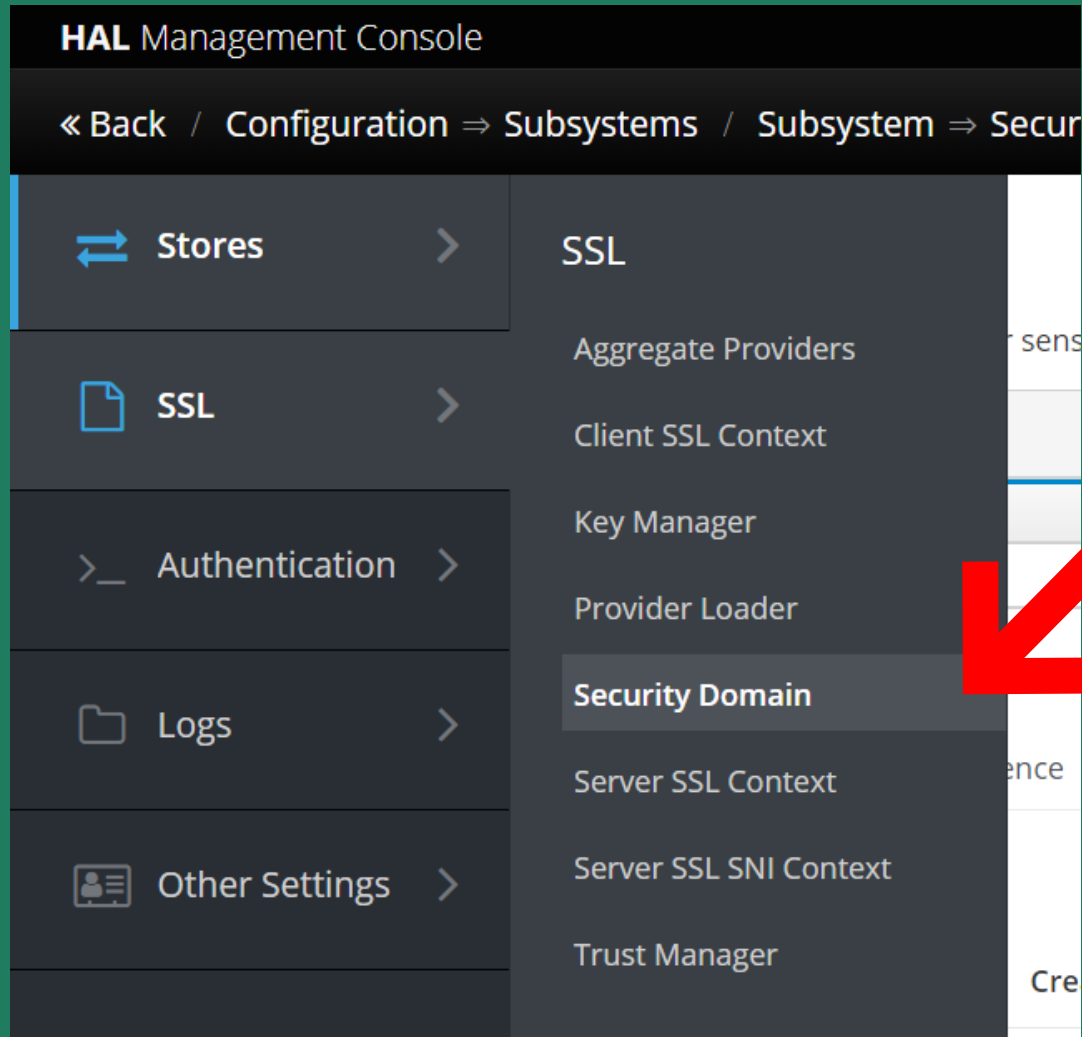
Cancel Save

Passons à la création d'un domain:Configuration->Subsystem->Other Settings-> View

The screenshot displays the HAL Management Console interface. The top navigation bar includes 'Homepage', 'Deployments', 'Configuration' (selected), 'Runtime', 'Patching', and 'Access Control'. The 'Configuration' section is expanded, showing a list of subsystems on the left and a list of settings on the right. A red arrow points to the 'Other Settings' link in the 'Settings' column, which has a 'View' button next to it.

Configuration	Subsystem (32)	Settings
Subsystems >	<i>Filter by: name or subtitle</i>	<i>Filter</i>
Interfaces >	Mail >	Global Settings
Socket Bindings >	Metrics	Factories / Transformers
Paths	MicroProfile JWT Smallrye	Mappers / Decoders
System Properties	Microprofile Config Smallrye	Other Settings View
	Naming JNDI	Security Realms
	Remoting	
	Request Controller	
	Resource Adapters >	
	Security Elytron >	

Puis SSL-> Security Domain



On click sur Add pour créer un nouveau domain.

On lui associe un realm par default.

On peut par la suite via le bouton Realm, rajouter d'autre realm à notre domain.

Add Security Domain ✕

[? Help](#)

Name *

monjdbcdomain

Default Realm *

jdbcrealm

▼

Required fields are marked *

Cancel

Add

Selectionnez notre domain et editez ses paramètres: nous allons utiliser les gestionnaires de permissions et de roles d'elytron par défaut.

Il est possible par la suite de les créer soit même.

monjdbcdomain

Realms

« < 1 of 1 > »

[? Help](#)

Default Realm

jdbcrealm

Evidence Decoder

Outflow Anonymous

🔗

OFF

Outflow Security Domains

Press .: to add new items and .x to remove them.

Permission Mapper

default-permission-mapper

Post Realm Principal Transformer

Pre Realm Principal Transformer

Principal Decoder

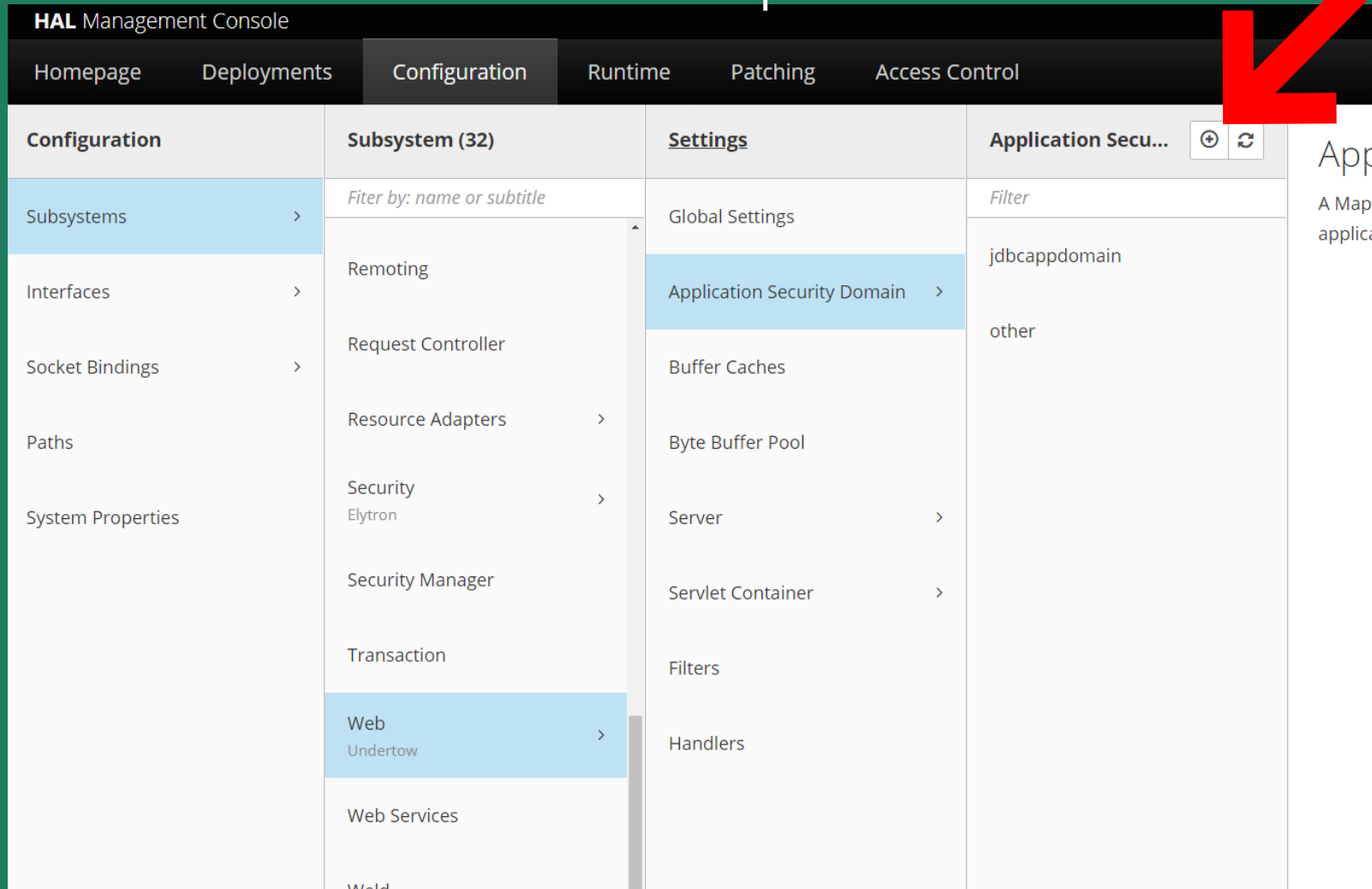
Realm Mapper

Role Decoder

groups-to-roles

3eme partie nous allons créer notre AppDomain.

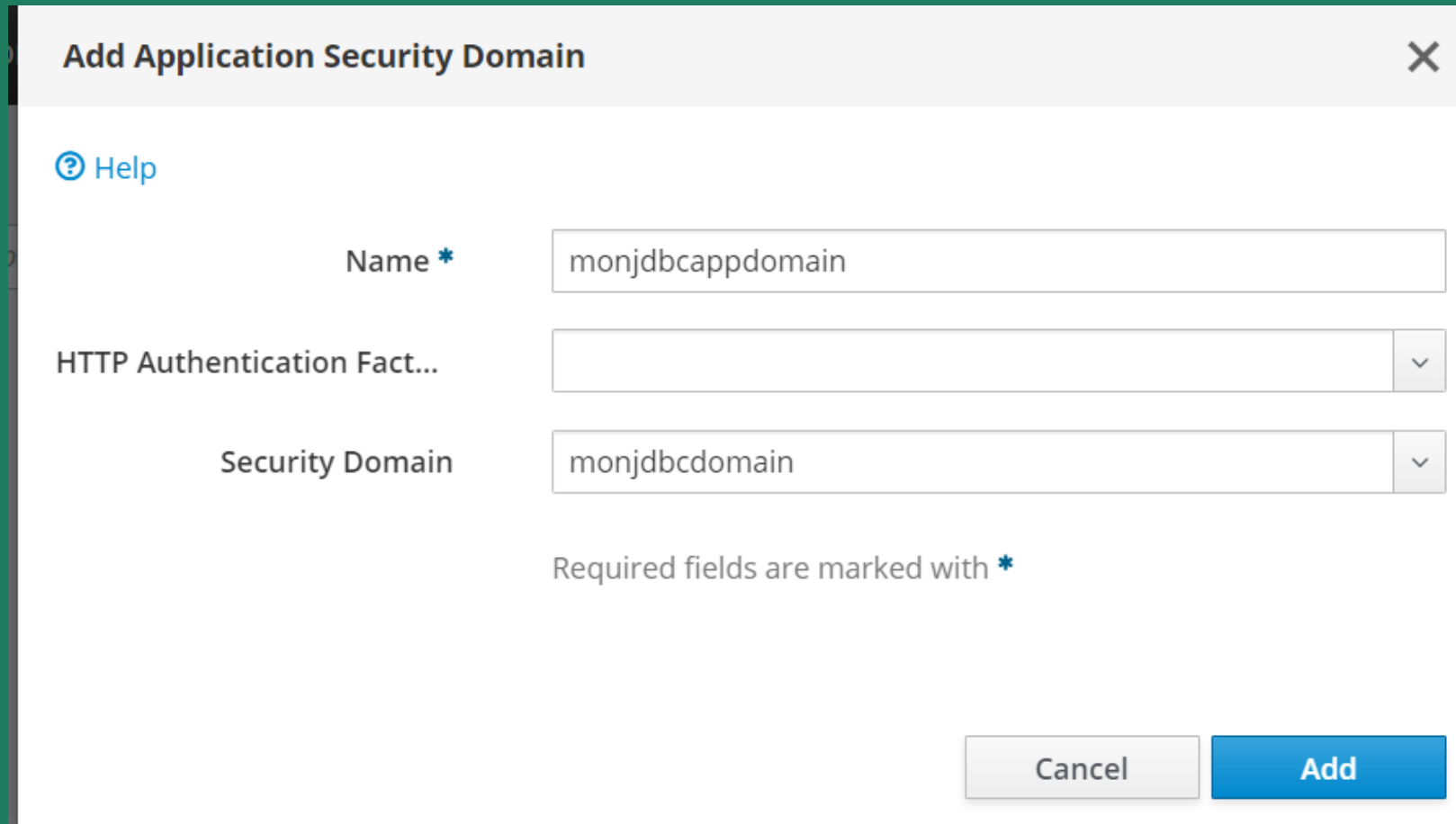
D'abord dans la partie Web:



The screenshot shows the HAL Management Console interface. The top navigation bar includes 'Homepage', 'Deployments', 'Configuration' (selected), 'Runtime', 'Patching', and 'Access Control'. A large red arrow points to the 'Configuration' tab. The main content area is divided into four columns: 'Configuration', 'Subsystem (32)', 'Settings', and 'Application Secu...'. The 'Configuration' column lists 'Subsystems', 'Interfaces', 'Socket Bindings', 'Paths', and 'System Properties'. The 'Subsystem (32)' column has a search filter and lists 'Remoting', 'Request Controller', 'Resource Adapters', 'Security Elytron', 'Security Manager', 'Transaction', 'Web Undertow' (selected), and 'Web Services'. The 'Settings' column lists 'Global Settings', 'Application Security Domain' (selected), 'Buffer Caches', 'Byte Buffer Pool', 'Server', 'Servlet Container', 'Filters', and 'Handlers'. The 'Application Secu...' column has a search filter and lists 'jdbcappdomain' and 'other'.

Configuration	Subsystem (32)	Settings	Application Secu...
Subsystems >	<i>Filter by: name or subtitle</i>	Global Settings	<i>Filter</i>
Interfaces >	Remoting	Application Security Domain >	jdbcappdomain
Socket Bindings >	Request Controller	Buffer Caches	other
Paths	Resource Adapters >	Byte Buffer Pool	
System Properties	Security Elytron >	Server >	
	Security Manager	Servlet Container >	
	Transaction	Filters	
	Web Undertow >	Handlers	
	Web Services		

3eme partie nous allons créer notre AppDomain.
D'abord dans la partie Web:



The screenshot shows a dialog box titled "Add Application Security Domain" with a close button (X) in the top right corner. Inside the dialog, there is a "Help" link (question mark icon) on the left. The main area contains three input fields: "Name *" with the value "monjdbcappdomain", "HTTP Authentication Fact..." which is empty, and "Security Domain" with the value "monjdbcdomain". Both the "Name" and "Security Domain" fields have an asterisk indicating they are required. At the bottom right, there are two buttons: "Cancel" and "Add". A note at the bottom states "Required fields are marked with *".

Add Application Security Domain

[Help](#)

Name * monjdbcappdomain

HTTP Authentication Fact...

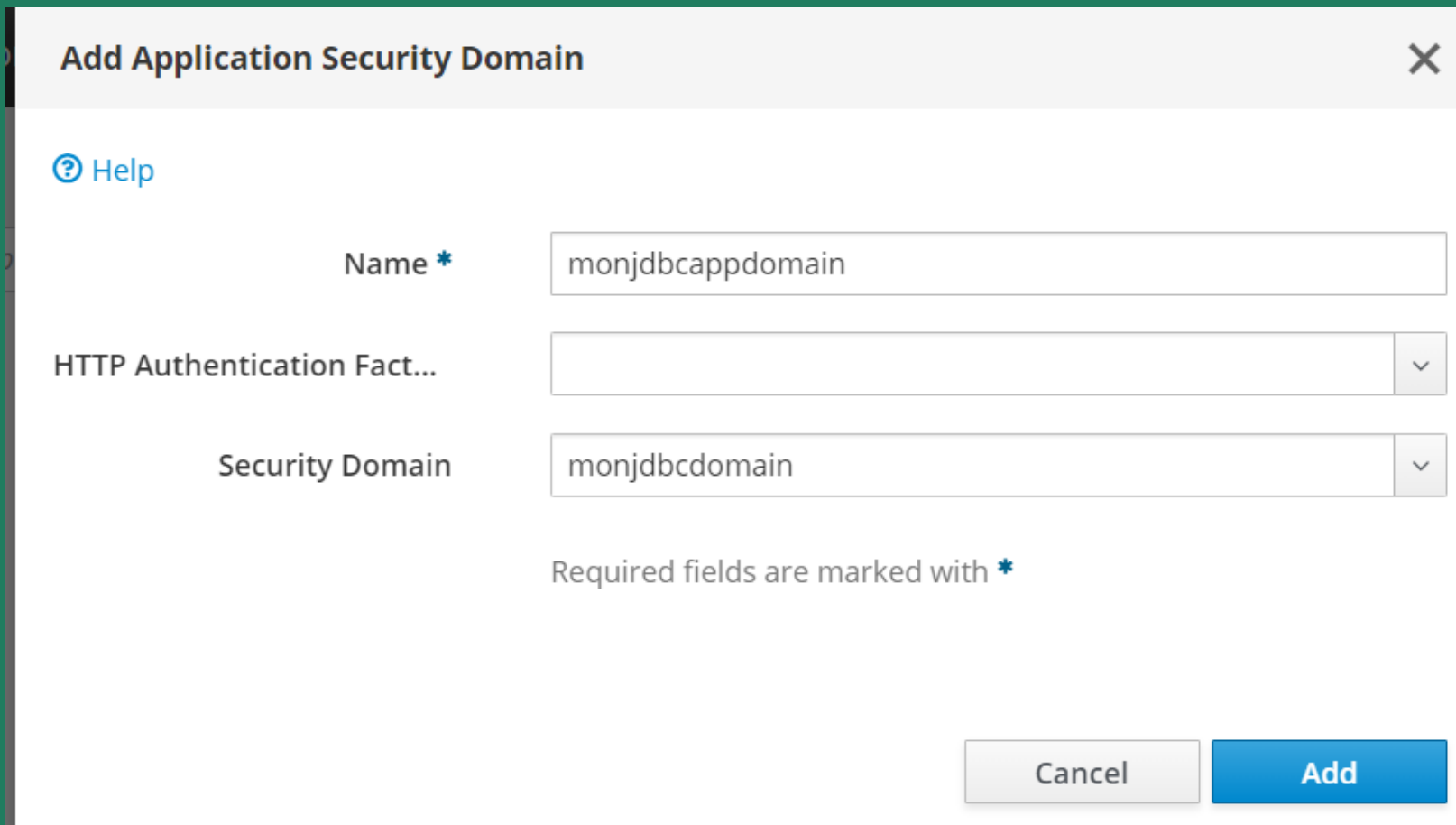
Security Domain monjdbcdomain

Required fields are marked with *

Cancel Add

On peut ici choisir de forcer les applications à une authentication http (necessite de créer une http authentication factory via le menu Security->Factory).

Ou alors juste lié l'AppDomain au domain et laissez les applications choisir leur mode d'authentification



The screenshot shows a dialog box titled "Add Application Security Domain" with a close button (X) in the top right corner. Inside the dialog, there is a "Help" link (question mark icon) in the top left. The main content area contains three labeled input fields: "Name *" with the value "monjdbcapppdomain", "HTTP Authentication Fact..." which is an empty dropdown menu, and "Security Domain" with the value "monjdbcdomain". A note at the bottom states "Required fields are marked *". At the bottom right, there are two buttons: "Cancel" and "Add".

Add Application Security Domain

[Help](#)


Name *

HTTP Authentication Fact...

Security Domain

Required fields are marked *

La 2eme partie de la configuration de l'AppDomain se trouve dans le menu EJB-> View

Homepage	Deployments	Configuration	Runtime
Configuration		<u>Subsystem (32)</u>	
Subsystems >		Filter by: name or subtitle	
Interfaces >		Deployment Scanners	
Socket Bindings >		Discovery	
Paths		Distributable Web 	
System Properties		EE	
		EJB View	

Puis SecurityDomain -> Add

The screenshot displays the HAL Management Console interface. On the left, a sidebar contains navigation links: Container, Bean Pool, State Management, Services, MDB Delivery, and Security Domain (which is highlighted). The main area shows the 'Add Application Security Domain' dialog box. This dialog has a title bar with a close button (X) and a 'Help' link. It contains two required fields: 'Name *' with the value 'monjdbcappdomain' and 'Security Domain *' with the value 'monjdbcdomain'. A note states 'Required fields are marked with *'. At the bottom right of the dialog are 'Cancel' and 'Add' buttons. The background shows a table with columns 'Name' and 'Security Domain', and rows for 'jdbcappdomain' and 'other'. Below the table, there are configuration options: 'Enable JACC' (false) and 'Legacy Compliant Principal Propagati...' (true).

HAL Management Console

« Back / Configuration ⇒ Subsystems / Subsys

Container >

Bean Pool

State Management

Services >

MDB Delivery

Security Domain

Application

A mapping from a sec

Name ^

jdbcappdomain

other

Help

Enable JACC false

Legacy Compliant Principal Propagati... true

Security Domain

Add Application Security Domain

Help


Name * monjdbcappdomain

Security Domain * monjdbcdomain

Required fields are marked with *

Cancel Add

Redémarrez votre serveur



HAL Management Console Reload Required

« Back / Configuration ⇒ Subsystems / Subsystem ⇒ EJB ▾

Container >

Bean Pool

State Management

Services >

MDB Delivery

Security Domain

Application Security Domain

A mapping from a security domain referenced in a deployed application

Showing 1 to 3 of 3 Items

Name ^
jdbccappdomain
monjdbappdomain
other

« < 1 of 1 »

[Help](#)

Enable JACC *false*

Legacy Compliant Principal Propagati... *true*

Alerts:

- The server configuration has changed. [Reload](#)
- Application Security Domain **monjdbappdomain** successfully added.

La configuration coté serveur est terminée.

A partir de ce moment, pour chaque application qu'on souhaite sécurisée via Elytron, on va fournir le nom de l'AppDomain.

D'abord via le fichier jboss-web.xml (dossier WEB-INF)

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <jboss-web xmlns="http://www.jboss.com/xml/ns/javaee"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="
5     http://www.jboss.com/xml/ns/javaee
6     http://www.jboss.org/j2ee/schema/jboss-web_5_1.xsd">
7   <security-domain>monjdbcappdomain</security-domain>
8
9
10 </jboss-web>
```

Puis via la configuration web.xml (plus obligatoire sur wildfly 25). C'est aussi via le web.xml que vous allez spécifier à l'application: Quel chemin est sécurisé , quel rôle à le droit de se connecter.

```
1 <security-constraint>
2
3   <web-resource-collection>
4
5     <web-resource-name>secure</web-resource-name>
6
7     <url-pattern>/*</url-pattern>
8
9   </web-resource-collection>
10
11  <auth-constraint>
12
13    <role-name>*</role-name>
14
15  </auth-constraint>
16
17 </security-constraint>
```

Ici toute l'application est sécurisée (/*), et tous les rôles peuvent se connecter une fois authentifié

On définit l'ensemble des rôles via les balises security role

```
1  <security-role>
2
3    <description>Le role admin/description>
4
5    <role-name>Admin</role-name>
6
7  </security-role>
8
9    <security-role>
10
11    <description>le role modérateur</description>
12
13    <role-name>Modo</role-name>
14
15  </security-role>
```

On peut aussi préciser la méthode d'authentification:

```
1 <login-config>
2
3   <auth-method>BASIC</auth-method>
4
5   <realm-name>monjdbcappdomain</realm-name>
6
7 </login-config>
```

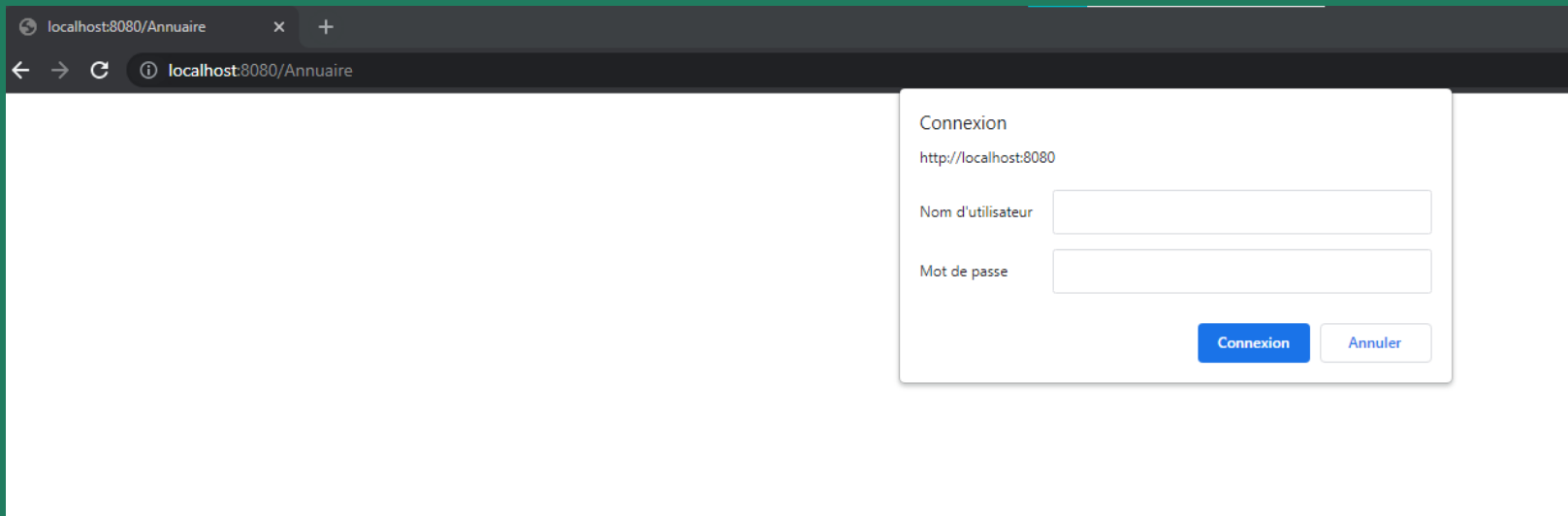

Web.xml complet

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/x
3   <display-name>Annuaire</display-name>
4   <session-config>
5     <session-timeout>30</session-timeout>
6   </session-config>
7   <mime-mapping>
8     <extension>ico</extension>
9     <mime-type>image/x-icon</mime-type>
10  </mime-mapping>
11  <welcome-file-list>
12    <welcome-file>/index.html</welcome-file>
13  </welcome-file-list>
14
15
16
17  <security-constraint>
18
19    <web-resource-collection>
20
21      <web-resource-name>secure</web-resource-name>
22
23      <url-pattern>/*</url-pattern>
24
25    </web-resource-collection>
26
27    <auth-constraint>
28
29      <role-name>*</role-name>
30
31    </auth-constraint>
```

Web.xml complet

```
1
2
3     </auth-constraint>
4
5 </security-constraint>
6
7 <security-role>
8
9     <description>The role that is required to log in to /secure/*</description>
10
11     <role-name>Admin</role-name>
12
13 </security-role>
14
15 <login-config>
16
17     <auth-method>BASIC</auth-method>
18
19     <realm-name>monjdbcappdomain</realm-name>
20
21 </login-config>
22 </web-app>
23
```

L'accès à l'application est maintenant sécurisée:



L'accès à l'application est maintenant sécurisé:

