Memoria Tecnica

Componente personalizado de conexión de diferentes bases de datos

Instalacion

1. Copiar el archivo gob.inap.identity.custom-1.0.0.jar a la carpeta ${CARBON\_HOME}/repository/components/lib
2. Agregar las siguiente configuración al archivo user-mgt.xml dentro de la carpeta ${CARBON\_HOME}/repository/conf:

<UserStoreManager class="gob.inap.identity.custom.CustomJDBCUserStoreManager">

<Property name="ReadOnly">false</Property>

<Property name="MaxUserNameListLength">100</Property>

<Property name="IsEmailUserName">false</Property>

<Property name="DomainCalculation">default</Property>

<Property name="PasswordDigest">SHA-256</Property>

<Property name="StoreSaltedPassword">true</Property>

<Property name="UserNameUniqueAcrossTenants">false</Property>

<Property name="PasswordJavaRegEx">[\S]{5,30}$</Property>

<Property name="PasswordJavaScriptRegEx">[\\S]{5,30}</Property>

<Property name="UsernameJavaRegEx">^[^~!@#$;%^\*+={}\\|\\\\&lt;&gt;]{3,30}$</Property>

<Property name="UsernameJavaScriptRegEx">[\\S]{3,30}</Property>

<Property name="RolenameJavaRegEx">^[^~!@#$;%^\*+={}\\|\\\\&lt;&gt;]{3,30}$</Property>

<Property name="RolenameJavaScriptRegEx">[\\S]{3,30}</Property>

<Property name="UserRolesCacheEnabled">true</Property>

</UserStoreManager>

1. Crear la carpeta ${CARBON\_HOME}/repository/conf/userStores
2. Crear dentro de la carpeta archivos de propiedades para cada base de datos sobre la que se realizaran búsqueda de usuarios. Debe de contener las siguientes propiedades:
   1. userstore.username: usuario de conexión a la base de datos.
   2. userstore.password: contraseña del usuario de conexión a la base de datos.
   3. userstore.url: url de conexión(JDBC) a la base de datos
   4. userstore.driver: Clase de driver de conexión a la base de datos
   5. userstore.maxActive: máximo numero de conexiones activas a las base de datos.
   6. userstore.minIdle: el minimo numero de conexiones dentro del pool antes de que se cree una nueva conexion
   7. userstore.maxWait: el numero de milisegundos que se esperara el resultado de una consulta.
   8. userstore.selectUser: La sentencia SQL que se utilizara para verificar la validez de un usuario. Debe de regresar un numero mayor a cero en caso de ser valido y menor o igual a cero en caso contrario. Debe de recibir dos parámetros, el primero debe de ser el usuario y el segundo la contraseña.
   9. userstore.algorithm: el algoritmo de reducción que se utilizo para almacenar la contraseña del usuario.

Ejemplo de archivo de configuración

userstore.username=root

userstore.password=root

userstore.url=jdbc:mysql://localhost:3306/houndqa

userstore.driver=com.mysql.jdbc.Driver

userstore.maxActive=50

userstore.minIdle=5

userstore.maxWait=60000

userstore.selectUser=Select count(\*) from usuarios where username=? and password=?

userstore.algorithm=SHA-256