BeanManagedConnection

The `<BeanManagedConnection>` element specifies an externally defined bean which defines a DataSource.

This is the most effective way to control connection properties, implement robust pooling support, and share connections across connectors.

Schema Name and Location

This element is defined by the `urn:mace:shibboleth:2.0:resolver` schema, which is located at http://shibboleth.net/schema/idp/shibboleth-attribute-resolver.xsd.

Example

The example illustrates a data source using the DBCP pooling library (version 1.4), which is a well-tested option.

```
<BeanManagedConnection>MyDataSource</BeanManagedConnection>
```

DataSource defined as a Spring Bean

```
<!-- Note that some of the settings below are defined as properties, which is optional. -->

<bean id="MyDataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close"
        p:driverClassName="%{datasource.driverClass}"
    p:url="%{datasource.jdbcUrl}" p:username="%{datasource.user}" p:password="%{datasource.password}"
    p:maxActive="10" p:maxIdle="5" p:maxWait="2000" p:testOnBorrow="true"
    p:validationQuery="select 1" p:validationQueryTimeout="5" />
```

Oracle Backend Example

```
<!-- Oracle database data source needs commons-dbcp-1.4.jar, commons-pool-1.x.jar, ojdbcx.jar in edit-webapp
/WEB-INF/lib/ -->
<!-- global.xml needs a new Oracle DB bean as in the previous example, but validationQuery needs to be modified
in "select 1 from dual" -->

<!-- Oracle Spring connection pooling data source configuration -->
<bean id="OracleDataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close"
    p:driverClassName="%{datasource.driverClass}"
    p:url="%{datasource.jdbcUrl}" p:username="%{datasource.user}" p:password="%{datasource.password}"
    p:maxActive="10" p:maxIdle="5" p:maxWait="2000" p:testOnBorrow="true"
    p:validationQuery="select 1 from dual" p:validationQueryTimeout="5" />

<!-- Attributes in idp.properties -->
datasource.driverClass = oracle.jdbc.OracleDriver
datasource.jdbcUrl = jdbc:oracle:thin:@<serverFQDN>:1521:<dbmame>
datasource.user = <USER>
datasource.password = <PASS>

<!-- example of attribute-resolver.xml portion for an attribute resolution using principalName as key-->
<DataConnector xsi:type="dc:RelationalDatabase" id="oracledbcf" >
    <BeanManagedConnection>OracleDataSource</dc:BeanManagedConnection>
    <QueryTemplate>
       <![CDATA[
             SELECT CF from DB.TABLE where USER_ID='$resolutionContext.principal'
       ]]>
    </QueryTemplate>
    <Column columnName="CF" attributeID="personalUniqueID_it_CF" />
</DataConnector>
```

Alternate Oracle Backend Example

```
<!-- Similar to the above, an Oracle database can also use commons-dbcp2.jar, commons-pool2.jar, ojdbc7.jar
with java 8 -->
<!-- global.xml needs a new Oracle DB bean as in the previous example, but validationQuery needs to be modified
in "select 1 from dual" -->

<!-- Oracle Spring connection pooling data source configuration -->
  <bean id="OracleDataSource" class="org.apache.commons.dbcp2.BasicDataSource" destroy-method="close"
    p:driverClassName="%{datasource.driverClass}"
    p:url="%{datasource.jdbcUrl}" p:username="%{datasource.user}" p:password="%{datasource.password}"
    p:initialSize="5" p:maxTotal="50" p:maxIdle="5" p:maxWaitMillis="2000" p:testOnBorrow="true"
    p:testWhileIdle="true" p:testOnReturn="true" p:timeBetweenEvictionRunsMillis="120000"
    p:minEvictableIdleTimeMillis="120000" p:validationQuery="select 1 from dual" p:validationQueryTimeout="4" />
```

Attributes

No attributes are defined.

Child Elements

No child elements are defined.