

## DataSource 설정방법

DataSource는 <datasources> 태그 엘리먼트 안에서 정의된다.

- <no-tx-datasource> - java.sql.Driver를 이용하되 JTA 트랜잭션을 사용하지 않는 DataSource
- <local-tx-datasource> - java.sql.Driver를 이용하되 2페이지 커밋을 지원하지 않는 DataSource
- <xa-tx-datasource> - java.sql.XADataSource를 이용한 2페이지 커밋을 지원하는 DataSource

### java.sql.Driver 를 이용한 파라미터 설정 방법

- <connection-uri > - URL 주소로 JDBC 드라이버 세팅하는 곳
- <driver-class> - java.sql.Driver 를 구현되어 있는 JDBC 드라이버 클래스
- <connection-property > - java.sql.Driver 에서 connection 생성을 위해 필요한 설정값  
예를 들자면  
<connection-property name="char.encoding">UTF-8</connection-property>

### javax.sql.XADataSource 를 이용한 파라미터 설정 방법 usage

- <xa-datasource-class> - XADataSource 를 구현한 클래스
- <xa-datasource-property> - XADataSource 에 필요한 설정값 (아래 예들)

```
<xa-datasource-property name="url">{XWAA11T1ME}</xa-datasource-property>
<xa-datasource-property name="{XWAA11T1ME}>myhost.mydomain.com/{xa-datasource-property name="PortNumber">1557</xa-datasource-property>
<xa-datasource-property name="DatabaseName">mydb</xa-datasource-property>
<xa-datasource-property name="ServerName">myserver</xa-datasource-property>
```

- <isSameRM-override-value> - 오라클에서 문제해결을 위해 필요한 세팅
- <track-connection-by-tx/> - 오라클에서 문제해결을 위해 필요한 세팅
- <no-tx-separate-pools/> - 오라클에서 문제해결을 위해 필요한 세팅

### 일반적인 DataSource 의 설정치

- <jni-name> - 바이팅되어지는 DataSource의 JNI 명
- <use-java-context> - JBoss 서버 VM 내에서 DataSource를 사용하기 위해 붙는 jndi 명의 앞에 붙여지는 prefix를 사용할 것인지 세팅하는 값. 디폴트는 true.
- <user-name> - DataBase 의 유저명
- <password> - DataBase 의 유저 패스워드
- <transaction-isolation> - DB Connection의 디폴트 transaction isolation 레벨. 명시 없으면 Database에서 제공하는 디폴트를 사용한다.
  - o TRANSACTION\_READ\_UNCOMMITTED
  - o TRANSACTION\_READ\_COMMITTED
  - o TRANSACTION\_REPEATABLE\_READ
  - o TRANSACTION\_SERIALIZABLE
  - o TRANSACTION\_NONE

- <new-connection-sql> - 새로운 connection 이 생성될 때 이 sql문이 실행된다. 대개는 connection schema를 세팅한다.
- <check-valid-connection-sql> - jndi에 주재하는 connection 의 validation

를 체크하기 위해 사용되어지는 sql 문 이다.

고 새로운 connection을 만든다. sql이 실패하는 경우 해당 connection 은 닫

- <valid-connection-checker-class-name> - 특정 벤더에서 제공하는 메카니즘을 사용하여 connection 이 유효한지 체크한다.
- <exception-sorter-class-name> - sql 에러가 발생시 심각성을 체크할 수 있는 벤더 의존적인 클래스
- <track-statements> - 사용자의 실수로 connection 을 닫지 않을 경우 경고 를 줄 수 있다. 디폴트는 false 이다.
- <prepared-statement-cache-size> - connection 에 캐쉬되는 prepared statements 의 갯수를 정할 수 있다. LRU 캐쉬형태로 지정된다. 디폴트는 0이다.
- <share-prepared-statements> - (a) 캐시되어 있는 prepared statement이 같은 트랜잭션내에서 같은 요청이 두번 실행된 경우 같은 결과를 반환되어 지게 된다.

- <query-timeout> - 쿼리 실행 시간의 최대치를 설정한다. (버전 4.0.3 이후)
- <metadata/typemapping> - conf/standardjbosscmp.xml (from JBoss4) 의 매핑 포인터로 사용된다.

(a) The purpose is to workaround questionable driver behavior where the driver applies auto-commit semantics to local transactions.

```
Connection c = dataSource.getConnection(); // auto-commit = false
PreparedStatement pstmt = c.prepareStatement(...);
ResultSet rs1 = pstmt.executeQuery();
PreparedStatement pstmt2 = c.prepareStatement(...);
ResultSet rs2 = pstmt2.executeQuery();
```

같은 prepared statements라고 가정할때 같은 드라이버에서 pstmt.executeQuery는 자동?

### 일반적인 DataSource 샘플

```
<<datasources>
<local-tx-datasource>
<jni-name>generic</jni-name>
<connection-uri>{jdbc: url for use with Driver class}</connection-uri>
<driver-class>{fully qualified class name of java.sql.Driver implementation}</driver-class>
<user-name>{user-name}
<password>{password}
<!-- you can include connection properties that will get passed in the DriverManager.getConnection(props) call-->
<!-- look at your driver docs to see what these might be -->
<connection-property name="char.encoding">UTF-8</connection-property>
<transaction-isolation>TRANSACTION_SERIALIZABLE</transaction-isolation>

<!--pooling parameters-->
<min-pool-size>5</min-pool-size>
<max-pool-size>100</max-pool-size>
<locking-timeout-millis>5000</locking-timeout-millis>
<idle-timeout-minutes>15</idle-timeout-minutes>
<!-- sql to call when connection is created
new-connection-sql>some arbitrary sql</new-connection-sql>
-->

<!-- sql to call on an existing pooled connection when it is obtained from pool
<check-valid-connection-sql>some arbitrary sql</check-valid-connection-sql>
-->

<set-tx-query-timeout/>
```

## DataSource 설정방법

```
query-timeout>query-timeout<!-- maximum of 3 minutes for queries -->

<!-- pooling criteria. USE AT MOST ONE-->
<!-- If you don't use JAAS login modules or explicit login
getConnection(usr, pw) but rely on user/pw specified above,
don't specify anything here -->
<!-- If you supply the usr/pw from a JAAS login module -->
<security-domain>MyRealm</security-domain>
<!-- if your app supplies the usr/pw explicitly getConnection(usr, pw) -->
<application-managed-security/>

<!-- Anonymous depends elements are copied verbatim into the ConnectionManager mbean cor
<depends>myapp.service=DataSourceServices</depends>

</local-tx-datasource>

<!-- you can include regular mbean configurations like this one -->
<mbean code="org.jboss.im.XidFactory"
name="jboss:service=XidFactory"
attribute name="Padi">true</attribute>
</mbean>

<!-- Here's an xa example -->
<xa-datasources>
<jndi-name>GenericXADataSource</jndi-name>
<xa-datasource-class>fully qualified name of class implementing javax.sql.XADataSource
<xa-datasource-property>"SomeProperty">SomePropertyValue</xa-datasource-property>
<xa-datasource-property name="SomeOtherProperty">SomeOtherValue</xa-datasource-property>
<user-name>xc/user-name>
<password>yc/password<
<transaction-isolation>TRANSACTION_SERIALIZABLE</transaction-isolation>

<!-- pooling parameters-->
<min-pool-size>9</min-pool-size>
<max-pool-size>100</max-pool-size>
<block-timeout-millis>5000</block-timeout-millis>
<idle-timeout-minutes>15</idle-timeout-minutes>
<!-- sql to call when connection is created
<new-connection-sql>some arbitrary sql</new-connection-sql>
-->
<!-- sql to call on an existing pooled connection when it is obtained from pool
<check-valid-connection-sql>some arbitrary sql</check-valid-connection-sql>
-->
<!-- pooling criteria. USE AT MOST ONE-->
<!-- If you don't use JAAS login modules or explicit login
getConnection(usr, pw) but rely on user/pw specified above,
don't specify anything here -->
<!-- If you supply the usr/pw from a JAAS login module -->
<security-domain>
<!-- if your app supplies the usr/pw explicitly getConnection(usr, pw) -->
<application-managed-security/>
</xa-datasource>
</datasources>
```