

Tomcat 5.5 JNDI Datasource の利用

[Tomcat]

<http://tomcat.apache.org/tomcat-5.5-doc/jndi-datasource-examples-howto.html>

Database Connection Pool(DBCP) の設定

<http://jakarta.apache.org/commons/dbcp/configuration.html>

インストール

DBCP は、Jakarta-Commons のコンポーネントに依存している。

- Jakarta-Commons DBCP
- Jakarta-Commons Collections
- Jakarta-Commons Pool

これらのライブラリは、ひとつの JAR ファイルとして、\$CATALINA_HOME/common/lib/naming-factory-dbc.jar に置かれている。

DB コネクションのリークを防ぐ

データベースコネクションプールは、データベースの接続を生成し、これを（新規作成するより効率がよいことから）再利用します。

これには、ひとつ問題があります。Web アプリケーションは、明示的に ResultSet や、Statement、Connection を閉じる必要があります。Web アプリケーションがこれらを閉じるのに失敗すると、これらは再利用されなくなってしまう。この状態がコネクションプールの " リーク " です。有効な接続を使い切ってしまうかもしれません。

DBCP では、このように不正に終了したコネクションを追跡し、復帰させることができます。

不正終了したコネクションを取り除き再利用するには、以下の属性を DBCP DataSource の設定に追加します。

```
removeAbandoned="true"
```

デフォルト値は false です

以下の設定をすることで、以下の秒数アイドル状態のコネクションを不正終了したとみなすことができます。

```
removeAbandonedTimeout="60"
```

デフォルト値は 300 秒です。

以下の設定を行うことで、スタックトレースログをとることができます。

```
logAbandoned="true"
```

デフォルト値は false です。

Oracle 8i、9i、10g

はじめに

- ・以前のバージョンの Oracle では、*.zip としてドライバが提供されていたが、Tomcat で利用できるのは、*.jar のみ。
- ・\$CATALINA_HOME/common/lib にインストールする。
- ・*.zip は *.jar にリネームする

Server.xml の設定

JNDI DataSource を \$CATALINA_HOME/conf/server.xml に記述する。

Context タグを Host タグの中に記述

```
<Context path="/myoracle" docBase="DBTest"
  debug="5" reloadable="true" crossContext="true">

  <!-- maxActive: Maximum number of dB connections in pool. Make sure you
  configure your mysqld max_connections large enough to handle
  all of your db connections. Set to 0 for no limit.
  -->

  <!-- maxIdle: Maximum number of idle dB connections to retain in pool.
  Set to -1 for no limit. See also the DBCP documentation on this
  and the minEvictableIdleTimeMillis configuration parameter.
  -->

  <!-- maxWait: Maximum time to wait for a dB connection to become available
  in ms, in this example 10 seconds. An Exception is thrown if
  this timeout is exceeded. Set to -1 to wait indefinitely.
  -->

  <!-- username and password: MySQL dB username and password for dB connections -->

  <!-- driverClassName: Class name for the old mm.mysql JDBC driver is
  org.gjt.mm.mysql.Driver - we recommend using Connector/J though.
  Class name for the official MySQL Connector/J driver is com.mysql.jdbc.Driver.
  -->

  <!-- url: The JDBC connection url for connecting to your MySQL dB.
  The autoReconnect=true argument to the url makes sure that the
  mm.mysql JDBC Driver will automatically reconnect if mysqld closed the
  connection. mysqld by default closes idle connections after 8 hours.
  -->

  <Resource name="jdbc/myoracle" auth="Container"
    type="javax.sql.DataSource" driverClassName="oracle.jdbc.OracleDriver"
    url="jdbc:oracle:thin:@127.0.0.1:1521:mysid"
    username="scott" password="tiger" maxActive="20" maxIdle="10"
    maxWait="-1"/>

</Context>
```

web.xml の設定

Web アプリケーションの Web.xml ファイルに、DTD の記述順に従い以下の記述を追記。

```
<resource-ref>
  <description>Oracle Datasource example</description>
  <res-ref-name>jdbc/myoracle</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

コード例

(1)

```
Context initContext = new InitialContext();
Context envContext = (Context) initContext.lookup("java:/comp/env");
DataSource ds = (DataSource) envContext.lookup("jdbc/myoracle");
Connection conn = ds.getConnection();
//etc.
```

(2)

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<sql:query var="rs" dataSource="jdbc/myoracle">
select id, foo, bar from testdata
</sql:query>

<html>
  <head>
    <title>DB Test</title>
  </head>
  <body>

    <h2>Results</h2>

    <c:forEach var="row" items="${rs.rows}">
      Foo ${row.foo}<br/>
      Bar ${row.bar}<br/>
    </c:forEach>

  </body>
</html>
```

DB2

- ・ Linux への Apache+Tomcat+DB2 インストール