

Derby

Integrating Apache Derby with Cocoon

add the driver to the WEB-INF/web.xml file

```
<init-param>
  <param-name>load-class</param-name>
  <param-value>
    org.apache.derby.jdbc.EmbeddedDriver
  </param-value>
</init-param>
```

download derby and copy the following jars into the WEB-INF/lib directory

- derby.jar
- derbynet.jar
- derbytools.jar

setup the datasource in the WEB-INF/cocoon.xconf

```
<datasources>
  <jdbc logger="core.datasources.personnel" name="personnel">
    <pool-controller max="10" min="5"/>
    <dburl>jdbc:derby:testdb:create=true</dburl>
    <user/>
    <password/>
  </jdbc>
</datasources>
```

use the datasource e.g. with ESQL

```
<?xml version="1.0"?>

<xsp:page language="java"
  xmlns:xsp="http://apache.org/xsp"
  xmlns:esql="http://apache.org/cocoon/SQL/v2">

  <page>
    <title>A Database Driven XSP Page</title>
    <resources>
      <resource type="file" href="../transform/sql-page.xml.sql">Schema</resource>
      <resource type="doc" href="userdocs/xsp/esql">ESQL</resource>
    </resources>

    <content>

      <esql:connection>
        <esql:pool>personnel</esql:pool>

      <!--
        <esql:execute-query>
          <esql:query>create table employee ( firstname varchar(20), lastname varchar(20) )</esql:query>
        </esql:execute-query>
      -->

      <esql:execute-query>
        <esql:query>delete from employee</esql:query>
      </esql:execute-query>

      <esql:execute-query>
        <esql:query>insert into employee (firstname, lastname) values ('Torsten1', 'Curdt')</esql:query>
      </esql:execute-query>
```

```

<esql:execute-query>
  <esql:query>insert into employee (firstname, lastname) values ('Torsten2', 'Curdt')</esql:query>
</esql:execute-query>
<esql:execute-query>
  <esql:query>insert into employee (firstname, lastname) values ('Torsten3', 'Curdt')</esql:query>
</esql:execute-query>
<esql:execute-query>
  <esql:query>insert into employee (firstname, lastname) values ('Torsten4', 'Curdt')</esql:query>
</esql:execute-query>

<esql:results/>

<para>
  Below you see all guys from the departments
</para>

<esql:execute-query>
  <esql:query>select * from employee</esql:query>
  <esql:results>
    <esql:row-results>
      <para><esql:get-string column="firstname"/></para>
    </esql:row-results>
  </esql:results>
</esql:execute-query>

<hr/>

<para>
  Below you see just some of the rows
</para>

<esql:execute-query>
  <esql:query>select * from employee</esql:query>
  <esql:use-limit-clause>jdbc</esql:use-limit-clause>
  <esql:skip-rows>1</esql:skip-rows>
  <esql:max-rows>2</esql:max-rows>
  <esql:results>
    <esql:row-results>
      <para><esql:get-string column="firstname"/></para>
    </esql:row-results>
    <esql:previous-results>
      We skipped some.
    </esql:previous-results>
    <esql:more-results>
      There are some more.
    </esql:more-results>
    <para>
      There is a total of <esql:row-count/> employees.
    </para>
  </esql:results>
</esql:execute-query>

</esql:connection>

</content>
</page>
</xsp:page>

```