

# XSPTransformCustomDate

If you are using <esql/> you may need to transform a custom date string to a valid Java Date object.

Here is an example:

```
<esql:query>
select * from reports where date = #<xsp-request:get-parameter name="date"/>#
</esql:query>
```

The actual date of the <esql:query/> has to be in the format e.g. 01/29/2003. Using the esql parameter will result in java code that looks something like the following:

```
PreparedStatement statement = conn.prepareStatement("select * from reports where date = ?");
statement.setDate(1, myDate);
```

... but if you are in e.g. Germany you are used to this date format: 29.01.2003.

**Step 1** Create a HTML-Form save it as date.html in your cocoon-webapp dir.

```
<html>
<head>
<meta name="author" content="ThorstenScherler">
<title>TransformCustomDate</title>
</head>
<body>
<p>Please press Submit.</p>
<form action="date.xml" method="GET">
<input type="text" name="date" value="29.01.2003"/>
<input type="submit" value="Submit"/>
</form>
</body>
</html>
```

On submit this will result in attaching **/date.xml?date=29.01.2003** to the uri where the html form was called from.

**Step 2** Create a file date.xsp and put this in your cocoon-webapp dir.

```

<xsp:page language="java" xmlns:xsp="http://apache.org/xsp">
<xsp:structure>
<xsp:include>java.util.*</xsp:include>
<xsp:include>java.text.*</xsp:include>
</xsp:structure>
<document>
<xsp:logic>
String datePattern = "dd.MM.yyyy";
SimpleDateFormat dateFormat = new SimpleDateFormat( datePattern );
String sDate_one = "";
    try
    {
        sDate_one=request.getParameter("date");
    }
    catch( ParseException e )
    {
        getLogger().error("XSP param error: ", e);
    }
Date date_one = null;
    try
    {
        date_one = dateFormat.parse( sDate_one );
    }
    catch( ParseException e )
    {
        getLogger().error("XSP date error: ", e);
    }
</xsp:logic>
<request>date.xml?sDate=<xsp:expr>date_one</xsp:expr>
</request>
<transformed>
<xsp:expr>date_one</xsp:expr>
</transformed>
</document>
</xsp:page>

```

**Step 3** put this in your cocoon-webapp sitemap.xmap

```

<map:pipeline>
<map:match pattern="date">
<map:read src="date.html"/>
</map:match>
<map:match pattern="date.xml">
<map:generate type="xsp" src="date.xsp"/>
<map:serialize type="xml"/>
</map:match>
</map:pipeline>

```

**Testing** Now request <http://localhost:8080/cocoon-webapp/date> and press the Submit-Button.

If everything works fine you will see a XML-page with a valid date. If not - please check the logs.

### Modifying the date-Format

```
String datePattern = "dd.MM.yyyy";
```

If you wish to modify the date-format please see [SimpleDateFormat Class Reference](#).

Giving you an idea of how to do it:

```
String datePattern = "MM/dd/yyyy";
```

expect 01/31/2003 as the string-format.

Please let me know if you had problems with it. Scherler