

Using LibSBML with Tomcat

We frequently get a request to provide information on how to use LibSBML with Tomcat. This document details how to use the LibSBML Java bindings with Tomcat. This procedure has been verified to work with:

- Tomcat 6.0.29
- Tomcat 7.0.5

on Windows (7, 64bit), Linux (Fedora Core 14, 64bit) and Mac OS X Snow Leopard (10.6.5).

Step 1: Installing Tomcat

Download the Tomcat version you would like to use from:

<http://tomcat.apache.org/> and ensure that it works correctly, perhaps by using the sample.war file as available from: <http://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/>

If all is set up correctly, you should see the following result when navigating to <http://localhost:8080/sample>

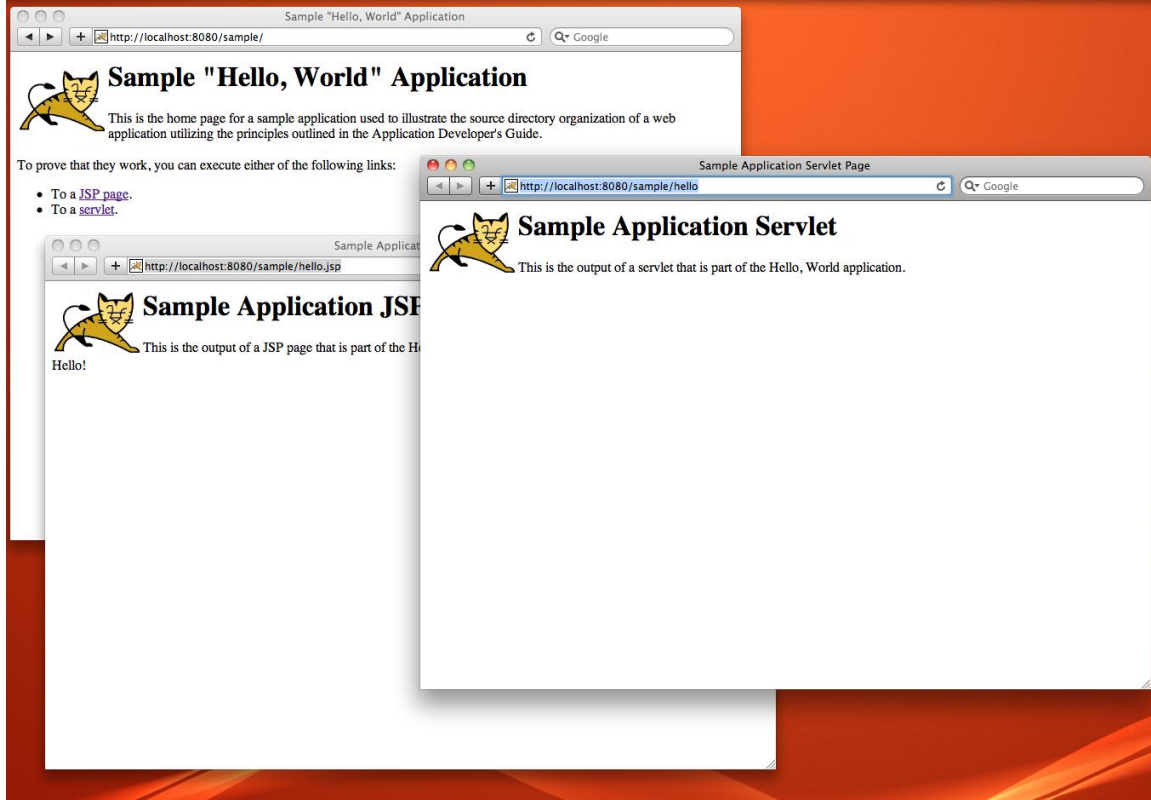


Figure 1: Sample Application working

To do this all that I did on my machine was to:

- Download tomcat:
 - `curl -L http://apache.mirrors.pair.com/tomcat/tomcat-7/v7.0.5-beta/bin/apache-tomcat-7.0.5.tar.gz -o apache-tomcat-7.0.5.tar.gz`
- Extract the archive:
 - `tar xzf apache-tomcat-7.0.5.tar.gz`
- Made all shell scripts executable
 - `cd apache-tomcat-7.0.5/bin/`
 - `chmod a+x *.sh`
- edit the `conf/tomcat-user.xml` file to include a manager user:
 - add the line

```
<user username="tomcat" password="tomcat"
      roles="manager,manager-gui,manager-status,admin-gui"/>
```

- start the webserver for testing by starting `bin/startup.sh`

Step 2: Add the LibSBML libraries to the tomcat libraries

Next we add the `libsbml javabinings` to the `apache-tomcat-<version>/lib` directory. For me that was:

Linux:

`libsbmlj.jar` as installed in `/usr/local/share/java/`
`libsbmlj.so` as installed in `/usr/local/lib`

(ensure that dependent libraries can be found when you type `ldd libsbmlj.so`)
(this assumes that you compiled and installed `libsbml 4.2.0` with the options
`./configure --with-java && make && sudo make install`)

OS X:

`libsbmlj.jar` as installed in `/usr/local/share/java/`
`libsbmlj.jnilib` as installed in `/usr/local/lib`

(rename `libsbmlj.jnilib` to `libsbmlj.dylib` (as that is what is going to be loaded on snow leopard))

(ensure that dependent libraries can be found when you type `otool -L libsbmlj.dylib`)

(this assumes that you compiled and installed `libsbml 4.2.0` with the options
`./configure --with-java && make && sudo make install`)

Windows

`sbmlj.jar` as installed in `c:\program files\SBML\libsbml-4.2.0-libxml2-x64\bindings\java`

sbmlj.dll as installed in c:\program files\SBML\libsbml-4.2.0-libxml2-x64\bindings\java
all *.dll files from c:\program files\SBML\libsbml-4.2.0-libxml2-x64\win64\bin

(this assumes that you have installed the libsbml-4.2.0 binary from:
<http://sourceforge.net/projects/sbml/files/libsbml/4.2.0/libSBML-4.2.0-win-libxml2-x64.exe/download>)
(the same will work for the 32bit build, by simply substituting x64 with x86 and win64 with win32 in the above)

Step 3: Ensure that these libraries are found by Tomcat

Next we make sure that these libraries are found by tomcat. For this we create a setenv script that puts the lib path into the environment variables:

Linux:

Create a setenv.sh file in the bin folder that includes the line:
export LD_LIBRARY_PATH=<fullpathToTomcatDir>/lib:\$LD_LIBRARY_PATH

(<fullpathToTomcatDir> needs to be replaced with the actual path)

OS X:

Create a setenv.sh file in the bin folder that includes the line:
export

DYLD_LIBRARY_PATH=<fullpathToTomcatDir>/lib:\$DYLD_LIBRARY_PATH

(<fullpathToTomcatDir> needs to be replaced with the actual path)

Windows:

Create a setenv.bat file in the bin folder that includes the line:
Set PATH=<fullpathToTomcatDir>/lib;%PATH%

(<fullpathToTomcatDir> needs to be replaced with the actual path)

Step 4: Write an initializer class that loads the native sbmlj library.

Next we write an initializer for loading the native library. This is necessary, as calling System.loadLibrary from within a servlet / JSP page would be called repeatedly and Java does not allow that. In my case the full listing of that initializer class is:

```

package mypackage;

import org.sbml.libsbml.libsbml;

public class Initializer {

    static
    {
        System.loadLibrary("sbmlj");
    }

    public static void Initialize()
    {
        System.out.println("Initialized!");
        System.out.println("Using LibSBML: " +
                           libsbml.getLibSBMLDottedVersion());
    }
}

```

Listing 1: The initializer code

And the first thing a JSP /servlet does is to call this initializer. (The actual code in the initializer is not important, so you could replace it with whatever makes sense for your application).

The corresponding class file is also to be placed in the tomcat/lib directory. (In the case for the above file that would be lib/mypackage/Initializer.class).

Step 5: Testing using LibSBML in a JSP page

That's it LibSBML is ready to be used in a jsp page. So let us try it.

As discussed before the JSP package will call the initializer once (best in the `jspInit` method), and then is free to use `libsbml` as it sees fit. The following listing call `libsbml`, to print the LibSBML version:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" import="org.sbml.libsbml.*"
import="mypackage.*"%>

<%!

public void jspInit()
{
    Initializer.Initialize();
}

%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>Insert title here</title>
</head>
<body>
    <%= new String("Hello from JSP!") %>
    <%= new String(libsbml.getLibSBMLDottedVersion()) %>
</body>
</html>
```

To test it, simply:

- create a directory <directory> in tomcat/webapps
- and save the listing into the directory <directory> as <jspfilename>
- start tomcat
- navigate to : <http://localhost:8080/<directory>/<jspfilename>>

Step 6: Using LibSBML from a servlet.

Similarly it is possible to use LibSBML in a servlet. In the listing below again the libsbml version is printed.

```
package mypackage;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.sbml.libsbml.*;
```

```

/**
 * Servlet implementation class Hello
 */
@WebServlet(description = "test program testing libsbml", urlPatterns = { "/Hello" })
public class Hello extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Hello() {
        super();

        // call libsbml initializer
        Initializer.Initialize();
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        handleRequest(request, response);
    }

    private void handleRequest(HttpServletRequest request,
        HttpServletResponse response) throws IOException {

        response.getOutputStream().println("Hello from servlet!");
        response.getOutputStream().println("Using libsbml version: " +
            libsbml.getLibSBMLDottedVersion());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        handleRequest(request, response);
    }
}

```

Downloading a canned version

If you would like to simply test the above, you could also download a canned version from:

<http://fbergmann.github.com/libsbml-tomcat-tutorial/>

it contains the full Tomcat 7.0.5 version with the libsbml 4.2.0 language bindings. Simply download the tar.gz file, extract, and adapt the setenv.sh script in the bin folder, or create a setenv.bat file for windows. (see Step 3: Ensure that these libraries are found by Tomcat).

Since the archive contains the libsbml bindings from OS X Snow Leopard, you want to replace them with the ones for your operating system (see Step 2: Add the LibSBML libraries to the tomcat libraries).