

How-to build a Struts application with XDoclet

Michael Krauße

August 5, 2005

Keywords

Java, XDoclet, Struts, JBoss, J2EE

Abstract

This article describes howto build a simple Struts based web-application with the help of XDoclet. At the end of this article you will have a simple Struts based web-application from which it is easy to start your own project.

Contents

1 Introduction	1
1.1 License	1
1.2 External Links	1
1.3 Conclusion	1
1.4 Contact	1
2 Requirements	2
3 Description of the example	2
4 Go through	2
4.1 BookAction	2
4.2 BookForm	2
4.3 build.xml	3
5 Prepare build process	4
5.1 Where are your XDoclet libs	4
5.2 Where is your JBoss	4
5.3 Customise other stuff	4
6 Build, deploy and run the application	4

1 Introduction

You find a pdf version of this file at http://www.m-krausse.de/articles/xdoclet_struts.pdf.

1.1 License

This article is published under a creative commons license. The version which is valid for this document you will find here <http://creativecommons.org/licenses/by-sa/2.0/de/>. In short: commercial use and modification is permitted if you publish your changes under the same license.

The source code I provide with this example is based on an other example which I found several months ago in the World Wide Web. Since I am not able to refind it, I distribute this source code under the BSD License.

1.2 External Links

I have no control over external websites to whom this article links to. It is possible that the website owner (re)move the linked content and you are not able to access the website. If you can not find a website, please inform me. Sometimes I can send you a snapshot of the website.

1.3 Conclusion

After understanding how XDoclet works this tool saves time to develop web applications because source code is written once and dependent description files are auto-generated. Some description files can not be auto-generated and still need editing by hand. But this description files change seldom and can mainly be used from the example files. Especially if you build your application from a prototype which results in frequently renaming classes and methods plus moving files XDoclet saves your time because XDoclet takes care that your description files are *synced* with your source code.

1.4 Contact

On <http://www.m-krausse.de/impressum.html> you get an information how to write me an email. I'm sorry that I can't provide my email adress here thank those spammers.

2 Requirements

To understand this article you need a basis knowledge of Struts[3], XDoclet[4], Apache Ant[1] and JBoss[2] because I will not explain how those technologies work. This article focuses on howto to combine those technologies to build and deploy a Struts web-application on the JBoss Application-server. Although I developed the examples under Debian Linux I believe that you should be able to get the example running on every platform which is supported by Sun's JDK. I used the following versions of the projects: JBoss 4.0.2, Ant 1.6.2, XDoclet 1.2.2, Struts 1.2.7.

If you want to build the example you need the sources which you can find at <http://www.m-krausse.de/j2ee/strutsexample1.zip>.

3 Description of the example

The example is a small web-application. It consists of a testJSP.jsp, where a user enters the name of an book author and the book's title, a BookAction class which is used to store the information, and a BookForm which is used to check whether the data was entered correctly, a savedOK.jsp which is shown to the user if the book was saved successfully.

4 Go through

Because you can load all source-files in your favourite editor I only print here those parts of the file which are relevant for creating struts description files with XDoclet. The three dots above and under a code-snippet show that the original source file continues in both directions.

4.1 BookAction

The lines above give a snapshot of the XDoclet Tags which where added to BookAction.java. The struts.action tag informs XDoclet that the following class inherits from org.apache.struts.action.Action and should be auto described in the deployment file *struts-config.xml*. The path attribute says that the class should be bounded to book. This means that whenever the user request a file book.do from the Application Server this class is responsible for handling this request. We do not add the do to book because all *.do files should be first served by the standard Struts Action Class, I explain it later. The request type is somehow redundant but anyway it is needed to say XDoclet the type of this Action class. Attention the complete Name, including the package name, has to be provided. If you mess it up your application will not

Example 4.1 com/tek271/test/BookAction.java

```
...
/**
 * @struts.action
 *   path="/book"
 *   validate="true"
 *   type="com.tek271.test.BookAction"
 *   name="bookForm"
 *   input="/testJsp.jsp"
 *
 * @struts.action-forward
 *   name="saved"
 *   path="/savedOk.jsp"
 *   redirect="true"
 */
public class BookAction extends ...
```

work. With name you say which form should be used to enter the data. This form is called by Struts to enter the user Data. The input attribute says where Struts can find the form in which the user can insert the data. The action.forward tag informs struts what have to be done if the data is filled successfully. After the data was saved the method execute add "saved" to a mapping context. Which means that from now on something with save has to be done. What has to be done we define here. At first we use the signal word save and then we say what to do. In our case we want to redirect the request to savedOk.jsp which means that savedOk.jsp is send to the user. It contains a message that the book's data have been saved.

4.2 BookForm

Example 4.2 src/com/tek271/test/BookForm.java

```
...
/**
 *
 * @struts.form
 *   name="bookForm"
 */
public class BookForm extends ...
```

The struts.form tag informs XDoclet that the following class is a Form Class with the name bookForm. Be careful to choose the same name you selected in BookAction. If both names differ Struts is not able to fill your request because it does not find the form you are talking about at BookAction.

Example 4.3 src/com/tek271/test/BookForm.java

```

...
/**
 * @struts.form-field
 *
 * @struts.validator
 *   type="required"
 **/
public void setTitle ...

```

The tags say XDoclet that the field title has to validated. Validated means that data has to be filled in here. You find this tags above the method setAuthor too. Take care that you place those tags above the setter-methods and not above the getter methods.

4.3 build.xml**Example 4.4** taskdef

```

...
<taskdef name="webdoclet"
  classname="xdoclet.modules.web.
  WebDocletTask">
  <classpath>
    <path refid="xdoclet.classpath"/>
  </classpath>
</taskdef>
...

```

This snippet demonstrates how to add a customised ant tag to your build.xml. You define the new tag's name and which class is responsible for doing the work if that tag is called. Take care that XDoclet.classpath points to the place where your XDoclet library (those xdoclet*.jar files) are situated.

Example 4.5

```

<webdoclet destdir="${basedir}/WEB-INF"
  force="${xdoclet.force}"
  mergedir="metadata">

  <fileset dir="${src.dir}">
<include name="**/*" />
  </fileset>
...
</webdoclet>

```

You started the webdoclet tag and control where the output should be written to. We decide to write it to ./WEB-INF directory. The mergedir attribute is required because some description files need some extra input which can not be auto-generated from source code by XDoclet.

Example 4.6 generate web.xml

```

...
<deploymentdescriptor validatexml="true"
  distributable="false"
  displayname="${app.name}"
  mergedir="metadata/web"
 />
..

```

The deployment-descriptor tag is responsible to create the /WEB-INF/web.xml file. Since XDoclet is not able to generate this file alone we give some help with the files you find in metadata/web. With servlets.xml we define that we can use the servlet org.apache.struts.action.ActionServlet under the name action in the web.xml deployment descriptor. Next we need a servlet-mappings.xml file so we can tell the web-container what should be done if the user request matches the pattern *.do. As you can see we call the servlet we named "action". So the request book.do is first redirected to ActionServlet and afterward redirected to our BookAction Class. Last we need a taglibs.xml file here we say that the web-container should use the struts-html.tdl tag-library which was provided by Struts Framework. Struts provided other tag-lib's if you need them add them here.

Example 4.7 generate WEB-INF/struts-config.xml

```

<strutsconfigxml validatexml="true"
  version="1.1"
  mergedir="metadata/struts"
 />

```

The tag strutsconfigxml creates WEB-INF/struts-config.xml. XDoclet should validate the generated xml-file. We provide the merge-files at metadata/struts. The only file which we have to edit by hand is struts-message-ressources.xml. We define the ressourcefile which struts should use. This ressourcefile is important if you want to localise your web-application, providing it in different languages.

5 Prepare build process

After preparing everything that XDoclet can automatic generate all needed files it is time to start a test run. But before you start you have to do some configurations. Sorry but I can't do everything for you. So let's start it's easy .

5.1 Where are your XDoclet libs

If you open the build.xml file you see in the top the property xdoclet.cp. You have to change the value so that it points to the place where you installed your xdoclet*.jar files. You can not build the example if ant is unable to find those jar files.

5.2 Where is your JBoss

To compile servlets the compiler needs the servlet.jar file. It is provided by your JBoss installation. Please enter the place of your JBoss Server here. Be careful to enter the complete path to ONE of the three different servers which are provided by JBoss. I used the JBOSSHOME/server/default directory. This variable is also used to decide where the strutsexample1.war should be deployed.

5.3 Customise other stuff

You are able to control the name-space under which this web-application is reachable on JBoss if you change the property app.name and you can change the name of war.name if you prefer an other name of

6 Build, deploy and run the application

To build and deploy the application just enter ant deploy. After this check whether JBoss is running and whether it has deployed your application. Use your favourite browser and surf to <http://localhost:8080/strutsexample1/testJsp.jsp> and voilà here your Struts application generated with the help of XDoclet is presented to you.

References

- [1] *Homepage of Apache Ant*
<http://ant.apache.org> 2005-08-02
- [2] *Homepage of JBoss Project*

<http://www.jboss.org> 2005-08-02

- [3] *Homepage of Jakarta Struts Project*
<http://struts.apache.org/> 2005-08-02

- [4] *Homepage of Jakarta Struts Project*
<http://xdoclet.sourceforge.net> 2005-08-02