StrutsFileDownload

A new DownloadAction was added in Struts 1.2.6 (see the JavaDoc). This page is to show how to use it and has the following structure:

💡 FrankZ: Here is the sample webapp... downloadapp.zip Please note that this is my first ever attempt at editing a Wiki entry, so if I screwed anything up, I apologize in advance! Lastly, I skimmed through the content of this entry and didn't see anything blatantly wrong. I will try and proof it more thoroughly as time allows, but for now, in addition to the sample app, it should get anyone going in the right direction.

⚠️ niallp: Since I haven't actually used this class myself, I'm hoping that Martin, Frank or anyone else involved in the discussion/creation of this would take a look here to see if its OK.

💡 Details from this thread: Mail Archive - Note that the link to the sample app at omnytex.com referenced in this thread is no longer valid. The sample app is attached to this Wiki page only, so download it from here downloadapp.zip

Implementing a DownloadAction

You need to extend org.apache.struts.actions.DownloadAction and implement the getStreamInfo() method. Optionally you can also override the getBufferSize() method if you require a different buffer size from the default.

Implement the getStreamInfo() Method

The getStreamInfo() method returns a StreamInfo object - which is an inner interface of the DownloadAction. The DownloadAction provides two concrete implementations (static inner classes) of the StreamInfo interface:

- FileStreamInfo - Simplifies downloading of a file from disk - need to pass a java.io.File object to the constructor along with the content type.
- ResourceStreamInfo - simplifies downloading of a web application resource - need to pass the ServletContext, path and content type to its constructor.

In the examples below, I have also provided a Byte array implementation of the StreamInfo interface.

Implement the getBufferSize() Method

The DownloadAction, by default, returns a buffer size of 4096. Optionally, this may be overriden to customize the size of the buffer used to transfer the file.

Examples

Below are three examples:

- using a File
- using a web application resource
- using a byte array.

FileStreamInfo Example

Example of using the DownloadAction with a file. This example picks up the file name from the *parameter* attribute in the strust-config.xml action mapping.

```
import java.io.File;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.actions.DownloadAction;

public class ExampleFileDownload extends DownloadAction{

    protected StreamInfo getStreamInfo(ActionMapping mapping,
                                       ActionForm form,
                                       HttpServletRequest request,
                                       HttpServletResponse response)
            throws Exception {

        // Download a "pdf" file - gets the file name from the
        // Action Mapping's parameter
        String contentType = "application/pdf";
        File file          = new File(mapping.getParameter());

        return new FileStreamInfo(contentType, file);

    }

}
```

ResourceStreamInfo Example

Example of using the DownloadAction with a web application resource. This example picks up the web application resource path from the *parameter* attribute in the strust-config.xml action mapping.

```
import javax.servlet.ServletContext;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.actions.DownloadAction;

public class ExampleResourceDownload extends DownloadAction {

    protected StreamInfo getStreamInfo(ActionMapping mapping,
                                       ActionForm form,
                                       HttpServletRequest request,
                                       HttpServletResponse response)
            throws Exception {

        // Download a "jpeg" file - gets the file name from the
        // Action Mapping's parameter
        String contentType         = "image/jpeg";
        String path                = mapping.getParameter();
        ServletContext application = servlet.getServletContext();


        return new ResourceStreamInfo(contentType, application, path);

    }

}
```

Byte Array Example

Example of using the DownloadAction with a Byte Array.

This example creates a ByteArrayStreamInfo inner class which implements the StreamInfo interface.

(niallp: IMO we should include this ByteArrayStreamInfo in the implementation)

```
import java.io.IOException;
import java.io.InputStream;
import java.io.ByteArrayInputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.actions.DownloadAction;

public class ExampleByteArrayDownload extends DownloadAction {

    protected StreamInfo getStreamInfo(ActionMapping mapping,
                                       ActionForm form,
                                       HttpServletRequest request,
                                       HttpServletResponse response)
            throws Exception {

        // Download a "pdf" file
        String contentType = "application/pdf";
        byte[] myPdfBytes  = null;              // Get the bytes from somewhere

        return new ByteArrayStreamInfo(contentType, myPdfBytes);

    }

    protected class ByteArrayStreamInfo implements StreamInfo {

        protected String contentType;
        protected byte[] bytes;

        public ByteArrayStreamInfo(String contentType, byte[] bytes) {
            this.contentType = contentType;
            this.bytes = bytes;
        }

        public String getContentType() {
            return contentType;
        }

        public InputStream getInputStream() throws IOException {
            return new ByteArrayInputStream(bytes);
        }
    }
}
```

Using the DownloadAction in your Web Pages

The last bit of the puzzle is how do I use this action?

You need to do two things:

- As with any Struts action, you need to configure it in the struts-config.xml
- Use it on your web page like any other link to a file

So for example you might have something like the following in your struts-config.xml:

```
    <action path="/downloadMyPdfFile" type="myPackage.ExampleFileDownload" parameter="/foo/bar.pdf">
    <action path="/downloadMyImage"   type="myPackage.ExampleResourceDownload" parameter="/images/myImage.jpeg">
```

The on your jsp page, you might use these in the following way:

```
    <html:img action="downloadMyImage" alt="My Image" height="400" width="400"/>

    <html:link action="downloadMyPdfFile">Click Here to See the PDF</html:link>
```

Note you may need to set the nocache value to false in the controller section of your struts config file. Nocache set to true prevented me from being able to use Internet Explorer to download the file succesfully; Firefox and Safari worked fine.

```
    <controller contentType="text/html;charset=UTF-8" locale="true" nocache="false" />
```

Content Disposition

Setting the Content Disposition

DownloadAction doesn't cater for setting the content dispositon header. The easiest way is set it in the getStreamInfo() method, for example...

```
public class ExampleFileDownload extends DownloadAction{


    protected StreamInfo getStreamInfo(ActionMapping mapping,
                                       ActionForm form,
                                       HttpServletRequest request,
                                       HttpServletResponse response)
            throws Exception {

        // File Name
        String fileName = mapping.getParameter();

        // Set the content disposition
        response.setHeader("Content-disposition",
                           "attachment; filename=" + fileName);

        // Download a "pdf" file - gets the file name from the
        // Action Mapping's parameter
        String contentType = "application/pdf";
        File file          = new File(fileName);

        return new FileStreamInfo(contentType, file);

    }
}
```

Probably would want to play with the file name, to remove any path info first.

Content Disposition Values

You can set the content disposition to either download the file or open up in the browser:

- To open up in the browser: "inline; filename=myFile.pdf"
- To download: "attachment; filename=myFile.pdf"

Displaying images I always set the content disposition to the "inline" option.

NOTE: I'm not expert at this, just got it working by trial and error. I had problems and I'm seem to remember I had to play with setting the response headers to get it to work properly. There may also be browser issues - my app only has to work with IE. Anyone who knows more about this, feel free to amend this page.

FIX for IE: Content-Disposition Attachment Header Does Not Save File link