File Upload in jetty6

Tutorial on FileUpload in Jetty6

In Jetty5, MultiPartRequest was used for file uploading. In Jetty6, MultiPartRequest was changed to MultiPartFilter.

MultiPartFilter decodes the multipart/form-data stream sent by a HTML form that uses a file input item. A File object is then created and stored to the request as an attribute. The attribute name is equal to the name of the file input item of the form being submitted. Therefore if you have a file input item with a name of "userfile1", the attribute name for the File object is "userfile1".

Here is a quick tutorial for uploading files.

First, create the page with a form of enctype equal to multipart/form-data. File uploading won't work if you have a different enctype. Inside the form element, add an input file item.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<title>File Upload Sample</title>
</head>
<body>
<h1>File Upload Sample</h1>


<form action="fileupload" method="post"
enctype="multipart/form-data">

<table>
 <tbody>
  <tr>
   <td><input type="file" name="userfile1"
/></td>
  </tr>
  <tr>
```

```
   <td>
        <input type="submit" value="Submit"
/><input type="reset" value="Reset" />
   </td>
  </tr>
 </tbody>
</table>

</form>
```

```
</body>
</html>
```

Create a web.xml with filter and servlet mapping for /fileupload url because that's the action we specified in our form.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app
xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"
 xsi:schemaLocation="http://java.sun.com/xml
/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_
5.xsd"
 version="2.5">

 <display-name>File Upload
Sample</display-name>

 <servlet>
  <servlet-name>fileupload</servlet-name>

<servlet-class>com.exist.servlet.FileUpload<
/servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>fileupload</servlet-name>
```

```
  <url-pattern>/fileupload</url-pattern>
 </servlet-mapping>

 <filter>

<filter-name>fileuploadfilter</filter-name>

<filter-class>org.mortbay.servlet.MultiPartF
ilter</filter-class>
  <init-param>
   <param-name>deleteFiles</param-name>
   <param-value>true</param-value>
  </init-param>
 </filter>

 <filter-mapping>

<filter-name>fileuploadfilter</filter-name>
  <url-pattern>/fileupload</url-pattern>
 </filter-mapping>
```

```
</web-app>
```

Create a FileUpload servlet.

```
package com.exist.servlet;

import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import
javax.servlet.http.HttpServletRequest;
import
javax.servlet.http.HttpServletResponse;


@SuppressWarnings( "serial" )
public class FileUpload extends HttpServlet
{
    @SuppressWarnings( "unchecked" )
    protected void doGet( HttpServletRequest
req, HttpServletResponse resp ) throws
ServletException, IOException
    {
```

```
        PrintWriter outp = resp.getWriter();

        /*
         * if initParam deleteFiles = true
you can do either do step 1 or step 2
         *
         */
        // Step 1 --- modify the code inside
the for loop in your desired
        /*
        ArrayList files = (ArrayList)
req.getAttribute(
"org.mortbay.servlet.MultiPartFilter.files"
);
        for ( int x = 0; x < files.size();
x++ )
        {
            File file1 = (File) files.get( x
);
            File outputFile = new File(
"outputfile" + ( x + 1 ) );
            file1.renameTo( outputFile );
        }
         */

        StringBuffer buff = new
StringBuffer();

        File file1 = (File)
req.getAttribute( "userfile1" );
```

```
        if( file1 == null || !file1.exists()
)
        {
            buff.append( "File does not
exist" );
        }
        else if( file1.isDirectory())
        {
            buff.append( "File is a
directory" );
        }
        else
        {
            File outputFile = new File(
req.getParameter( "userfile1" ) );
            file1.renameTo( outputFile );
            buff.append( "File successfully
uploaded." );
        }

        outp.write( "<html>" );
        outp.write( "<head><title>FileUpload
page</title></head>" );
        outp.write( "<body>" );
        outp.write( "<h2>" + buff.toString()
+ "</h2>" );
        outp.write( "</body>" );
        outp.write( "</html>" );
    }

    protected void doPost(
```

```
HttpServletRequest req, HttpServletResponse
resp ) throws ServletException, IOException
    {
        doGet( req, resp );
```

```
        }

    }
```

The FileUpload servlet above simply stores the File being uploaded to the project target directory.

Simply modify the FileUpload servlet to your desired result. Simply bear in mind that the attribute name that you should get, should be the same as the file input item of the form.