MASTERTHEBOSS.COM

Home | JBoss AS | BPM | Projects | Web | SOA-Cloud | Eclipse | Java EE | Cool Stuff

WHAT IS JBOSS ?
Beginners can start learning in this section !

DEVELOP APPS
Start developing Java EE applications with JBoss AS

JBOSS BOOKS
Learn all about the application server

CONTACT US
Want to write for us? You would like to advertise here?

FOLLOW US
Follow us on Twitter! @mastertheboss

Primefaces file upload example

The PrimeFaces **File upload** component can be used to perform server side file uploads. Let's see in this quick tutorial how to achieve it in some simple steps.

The first thing you need is adding some libraries to your application. As a matter of fact, PrimeFaces file upload relies on Apache **commons-file-upload** and **commons-io libraries**. So dowload them and add them to your WEB-INF/lib path:

Here you can download Apache Commons io

Here you can download Apache Commons File Upload

As far as **web.xml** configuration is concerned you need to declare the PrimeFaces filter in order to enable file uploading

```
<filter>
```

```
  <filter-name>PrimeFaces FileUpload Filter</filter-name>
  <filter-class>org.primefaces.webapp.filter.FileUploadFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>PrimeFaces FileUpload Filter</filter-name>
  <servlet-name>Faces Servlet</servlet-name>
</filter-mapping>
```

Good, your configuration is completed. Let's see a concrete example:

```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:p="http://primefaces.org/ui">

  <h:head></h:head>

  <h:body>
     <h:form id="form" >

          <p:fileUpload fileUploadListener="#{fileUploadController.upload}"
6.allowTypes="/(\.|\/)(gif|jpe?g|png)$/" sizeLimit="100000" description="Select Images"/>

    </h:form>
  </h:body>
</html>
```

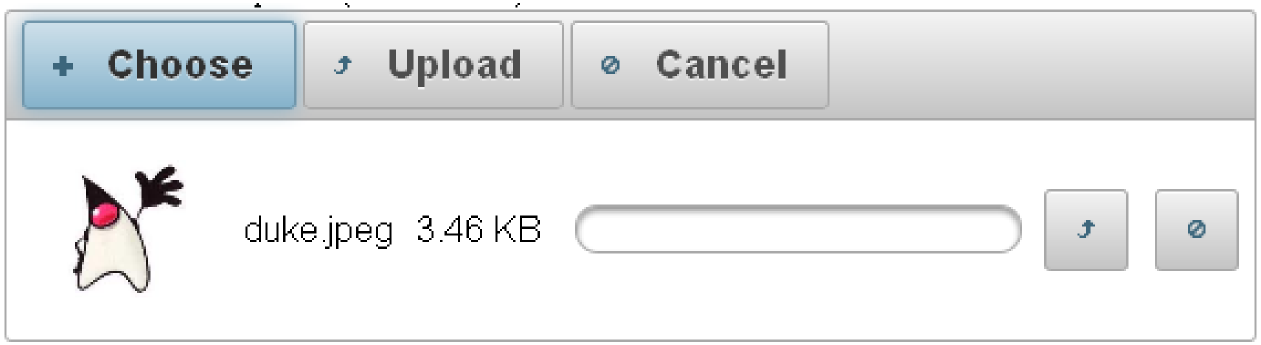
The tag does the job of sending the multipart stream to the server. In the bean declared (fileUploadController) a listener will fire which will let you handle the file and do what you want with it.

Here's the bean implementation:

```
package com.sample.bean;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.context.FacesContext;
```

```java

import org.primefaces.event.FileUploadEvent;

@ManagedBean(name="fileUploadController")
public class FileUploadController {
   private String destination="D:\\tmp\\";

    public void upload(FileUploadEvent event) {
        FacesMessage msg = new FacesMessage("Success! ", event.getFile().getFileName() + " is uploaded.");
        FacesContext.getCurrentInstance().addMessage(null, msg);
        // Do what you want with the file
        try {
            copyFile(event.getFile().getFileName(), event.getFile().getInputstream());
        } catch (IOException e) {
            e.printStackTrace();
        }

    }

    public void copyFile(String fileName, InputStream in) {
           try {


                // write the inputStream to a FileOutputStream
                OutputStream out = new FileOutputStream(new File(destination + fileName));

                int read = 0;
                byte[] bytes = new byte[1024];

                while ((read = in.read(bytes)) != -1) {
                    out.write(bytes, 0, read);
                }

                in.close();
                out.flush();
                out.close();

                System.out.println("New file created!");
                } catch (IOException e) {
                System.out.println(e.getMessage());
                }
    }
}
```

The upload method provides as parameter the FileUploadEvent from which you can retrieve the temporary file uploaded on the server. Once you have got it, use the stream to copy it or add it to a relational database.



As additional note, you can choose the path where the file is uploaded on the server by setting some additional parameters like uploadDirectory and thresholdSize

thresholdSize specifies the maximum file size in bytes to keep uploaded files in memory. If it exceeds this limit, it'll be temporarily written to disk.
uploadDirectory is the folder where to keep temporary files that exceed thresholdSize.

```
<filter>
  <filter-name>PrimeFaces FileUpload Filter</filter-name>
  <filter-class>org.primefaces.webapp.filter.FileUploadFilter</filter-class>
  <init-param>
    <param-name>thresholdSize</param-name>
    <param-value>51200</param-value>
  </init-param>
  <init-param>
    <param-name>uploadDirectory</param-name>
    <param-value>C:\etc</param-value>
  </init-param>
</filter>
```

You should however not rely on these attributes to define a stable storage of uploaded file. Always use IO Streams to collect the file and then store them in your favourite storage location<

Share +1 4 Tweet 0 Share Like 15 Send

Add a comment...

Shikha Baid · Works at APT Classes

where are the uploaded files stored?
please help me.

Reply · Like · December 18, 2012 at 8:11am

Aulia Yayan · 20 years old

can you help me. I'm already do it this step by step much time, but my prc
tutorial about fileupload in jsf+ primefaces, Please help me... :'(

Reply · Like · December 10, 2012 at 9:28am

Rezano Poltak Simanjuntak · University of Indonesia

wuidihhhhh, udah pake Jboss

Reply · Like · December 10, 2012 at 9:07pm

Anil Kurmi · S.p.jain gurukul khurai

nice tuts

Reply · Like · November 20, 2012 at 9:14am

Vanderson Silva · ★ Top Commenter · Programador at Giorgi & Santos

Pretty good article, congratulations! It helped me a lot!

Thank you!

Reply · 👍 2 · Like · November 10, 2012 at 11:26am

Akram Hamoudi · Works at Dtmode

thank you

Reply · Like · December 25, 2012 at 4:45pm

Facebook social plugin